Tivoli Workload Scheduler

Planung und Installation

Version 8.1

SH12-3010-00

Tivoli Workload Scheduler

Planung und Installation

Version 8.1

SH12-3010-00

Tivoli Workload Scheduler Planung und Installation Version 8.1

Copyrightvermerk

© Copyright IBM Corporation 2000, 2001. Alle Rechte vorbehalten. Kann nur gemäß der Softwarelizenzvereinbarung von Tivoli Systems bzw. IBM oder dem Anhang für Tivoli-Produkte der IBM Nutzungsbedingungen verwendet werden. Diese Veröffentlichung darf ohne vorherige schriftliche Genehmigung der IBM Corporation weder ganz noch in Auszügen auf irgendeine Weise - elektronisch, mechanisch, magnetisch, optisch, chemisch, manuell u. a. - vervielfältigt, übertragen, aufgezeichnet, auf einem Abrufsystem gespeichert oder in eine andere Computersprache übersetzt werden. Die IBM Corporation erteilt Ihnen eine begrenzte Berechtigung zur Erstellung einer Hardcopy oder sonstiger Vervielfältigungen in Form maschinenlesbarer Dokumentationen für die interne Verwendung, wobei jede Vervielfältigung den IBM Corporation Copyrightvermerk enthalten muss. Weitere das Copyright betreffende Rechte werden nur nach vorheriger schriftlicher Genehmigung durch die IBM Corporation gewährt. Die Veröffentlichung dient nicht für Produktionszwecke. Tivoli Systems übernimmt keine Haftung. **Die in diesem Dokument aufgeführten Beispiele sollen lediglich zur Veranschaulichung und zu keinem anderen Zweck dienen.**

Marken

IBM, Tivoli, das Tivoli-Logo, AIX, AS/400, BookManager, Dynix, OS/390 und Sequent sind in gewissen Ländern Marken der International Business Machines Corporation oder von Tivoli Systems Inc.

Intel ist eine eingetragene Marke der Intel Corporation.

Microsoft, Windows und Windows NT sind in gewissen Ländern Marken der Microsoft Corporation.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Java und alle auf Java basierenden Marken oder Logos sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

Bemerkungen

Hinweise auf Produkte, Programme oder Dienstleistungen von Tivoli Systems oder IBM in dieser Veröffentlichung bedeuten nicht, dass diese in allen Ländern, in denen Tivoli Systems oder IBM vertreten ist, angeboten werden. Hinweise auf diese Produkte, Programme oder Dienstleistungen bedeuten nicht, dass nur Produkte, Programme oder Dienstleistungen von Tivoli Systems oder IBM verwendet werden können. Anstelle der Produkte, Programme oder Dienstleistungen von Tivoli Systems oder IBM können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von Tivoli Systems oder IBM verletzen. Die Verantwortung für den Betrieb in Verbindung mit Fremdprodukten liegt beim Kunden, soweit solche Verbindungen nicht ausdrücklich von Tivoli Systems oder IBM bestätigt sind.

Für die in diesem Dokument beschriebenen Produkte und Verfahren kann es Patente oder Patentanmeldungen von Tivoli Systems oder IBM geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785, U.S.A. zu richten. Anfragen an obige Adresse müssen auf Englisch formuliert werden.

ISO 9001-Zertifizierung

Dieses Produkt wurde mit einem nach ISO 9001 zertifizierten Qualitätssystem entwickelt.

Die Zertifizierung erfolgte durch das Bureau Veritas Quality International (BVQI) (Zertifizierungsnummer BVQI - 92086 / A).

BVQI gehört weltweit zu den Marktführern im Bereich Qualitätszertifizierung und ist derzeit von mehr als 20 Zulassungsstellen anerkannt.

Inhaltsverzeichnis

Vorwort ix

Zielgruppe ix
Inhalt dieses Handbuchs ix
Veröffentlichungen x
Tivoli Workload Scheduler-Veröffentlichungen x
Zugehörige Veröffentlichungen x
Onlinebücher für Tivoli Workload Scheduler for z/OS xi
Online auf Veröffentlichungen zugreifen xi
Veröffentlichungen bestellen xi
Rückmeldung zu Veröffentlichungen xii
Plattformspezifische Informationen xii
Kontaktaufname mit der Kundenunterstützung xiii
In diesem Handbuch verwendete Konventionen xiii

Kapitel 1. Einführung 1

Neue Funktionen in dieser Version 1
Mehrere Kalender mit Feiertagen 1
Regel für arbeitsfreie Tage 2
Leistungsoptimierungen 2
Installationsoptimierungen 3
Linux-Unterstützung 3
Endpunkt-zu-Endpunkt-Zeitplanung 3
Integration in Tivoli Business Systems Manager 4
Gelöschte Funktionalitäten 4
Tivoli Workload Scheduler 4
Job Scheduling Console 6
Terminologie der Job Scheduling Console 7
Überlegungen zu Zeitzonen 8
Netzwerkplanung 9
Übersicht über ein Tivoli Workload Scheduler-Netzwerk 9
Funktionalität der Domänen 9
Auf eine Domäne begrenzte Verarbeitung 10
Überlegungen zum Netzwerk 10
Netzwerke mit einer Domäne 12
Netzwerke mit mehreren Domänen 13
Zu einem Sicherungsdomänenmanager wechseln 14
Erweiterte Datenbank und lange Objektnamen 14

Datenbankerstellung 15
Workstationnamen 15
Übersicht über die Installation 15
Produktgruppen 16
Komponentendatei 16
Die Komponentendatei anzeigen 17
Basisverzeichnis von Netman 17
Lokale Netman-Optionen 17
Nach der Installation 17

Kapitel 2. Unter Windows NT installieren 19

Systemvoraussetzungen 19
Für Endpunkt-zu-Endpunkt-Zeitplanung installieren 20
Aktualisierungen vorbereiten 20
Tivoli Workload Scheduler trennen und stoppen 20
Sicherungsdateien 21
Die Software installieren 21
Das Setup-Programm 21
Das Setup-Programm ausführen 21
Nach der Installation bzw. der Aktualisierung 28
Eine Aktualisierung abschließen 28
Die Variable PATH setzen 28
Verzeichnisse und Dienste von Tivoli Workload Scheduler 28
Der Tivoli Workload Scheduler-Dienst 28
Die dezentrale Verwaltung einrichten 29
Die Verzeichnisse des Masters gemeinsam benutzen 29
Tivoli Workload Scheduler-Parameter gemeinsam benutzen 29
Ein Verzeichnis gemeinsam benutzen 29
Die lokalen Optionen einstellen 30
Die lokalen Optionen auf dem Master einstellen 31
Das Tivoli Workload Scheduler-Konto manuell erstellen 31
Tivoli Workload Scheduler-Verwaltungstypen 31
Zentrale Verwaltung 31
Dezentrale Verwaltung 32
Tivoli Workload Scheduler deinstallieren 33
Das Tool ″Software″ in der Systemsteuerung von Windows NT verwenden 33

Kapitel 3. Unter UNIX installieren 35

Systemvoraussetzungen 35

Für Endpunkt-zu-Endpunkt-Zeitplanung installieren 35
Installationsverfahren 36
Die Tivoli Workload Scheduler-Steuerkomponente installieren 36
Konfigurationsschritte 38
Einen Hauptdomänenmanager nach der Installation konfigurieren 38
Einen fehlertoleranten Agenten nach der Installation konfigurieren 39
Tivoli Workload Scheduler aktualisieren 40
Vorbereitung: Tivoli Workload Scheduler stoppen 40
Sicherungskopien 40
Die Software aktualisieren und ″customize″ ausführen 41
Das Sicherheitsprofil aktualisieren 42
Ihre Dateien wiederherstellen 42
Tivoli Workload Scheduler erneut starten 42
Das Skript ″customize″ 43
Tivoli Workload Scheduler deinstallieren 46

Kapitel 4. Sicherheit definieren 49

Die Datei ″Security″ 49
Die Datei ″Security″ erstellen 49
Sicherheit in einem Netzwerk verwalten 49
Sicherheit für den Tivoli-Administrator definieren 49
Syntax der Datei ″Security″ 50
Benutzerdefinitionen 50
Syntax 50
Variablen 50
Reihenfolge für Benutzerqualifikation 58
Reihenfolge für Objektqualifikation 58
Von Tivoli bereitgestellte Variablen 59
Platzhalterzeichen 59
Der Superuser unter UNIX 59
Beispiel der Datei ″Security″ 60
Erklärung des Beispiels der Datei ″Security″ 63
Der Befehl ″dumpsec″ 65
Der Befehl ″makesec″ 66

Kapitel 5. Optionale Anpassungsoptionen 69

Globale Optionen 69
Globale Optionen setzen 69
Beispiel der Datei für globale Optionen 72

Die Weiterleitungsoptionen 73
Globale Optionen für MPE-Agenten setzen 74
Lokale Optionen 74
Lokale Optionen setzen 74
Beispiel der Datei für lokale Optionen 79
Lokale Optionen für Netman setzen 81
Optionen für dezentralisierte Verwaltung unter Windows NT setzen 81
Tivoli Workload Scheduler-Konsolnachrichten und -Eingabeaufforderungen 81
″syslog local″ unter UNIX setzen. 81
Der Befehl ″console″. 82
Den Produktionszyklus automatisieren 83
Das Jobnetz ″final″ anpassen 83
Das Jobnetz ″final″ hinzufügen 84
Einen Produktionszyklus starten. 84
Die Produktionsumgebung verwalten 84
Die Tagesstartzeit auswählen 84
Die Tagesstartzeit ändern 85
Einen Plan für zukünftige oder vergangene Tage erstellen 85
Die Konfigurationsskripts verwenden 86
Standardkonfigurationsskript - jobmanrc 86
Lokales Konfigurationsskript - $HOME/.jobmanrc 88

Anhang A. Zeitzonen und Prüfung 89

Zeitzonen 89
Die Zeitzonenfunktion aktivieren 90
Namen und Beschreibungen der Zeitzonen. 91
Prüfung. 92
Die Prüffunktion aktivieren 92
Das Format des Prüfprotokolls. 93
Prüfprotokoll-Header 97
Beispieleinträge für Prüfprotokolle. 97

Anhang B. Tivoli Workload Scheduler-Netzwerke 99

Definitionen 99
Tivoli Workload Scheduler for MPE 100
Netzwerkkommunikation 100
Netzwerkverbindungen. 101
Netzwerkoperation. 102
Netzwerkprozesse 102

Erweiterte Agenten 103
Erweiterte UNIX-Agenten 104
Die Methode für Zugriff auf lokalem UNIX-Agenten 104
Die Methode für Zugriff auf fernem UNIX-Agenten. 104
Die Produktion für erweiterte Agenten verwalten 105
Fehler beim Starten von Jobs auf einem erweiterten Agenten 105
Die Netman-Konfigurationsdatei 105
Die Netzwerk-IP-Adresse überprüfen 106
Systemkonfiguration (nur UNIX) 106
Fehlernachrichten/Warnungen. 107
Fehlerbehebung im Netzwerk. 108
Initialisierungsprobleme 108
Netzwerkverbindungsprobleme. 109
Anmerkungen 109
Einen Domänenmanager im Bereitschaftsmodus einrichten 109
Hauptdomänenmanager im Bereitschaftsmodus. 110
Hinweis zur Netzwerksicherheit 110
Ausfall eines Domänenmanagers 110
Einen Domänenmanager wechseln 111
Langfristiger Ausfall des Hauptdomänenmanagers 111

Anhang C. Netzwerkbetrieb mit MPE. 113

Überlegungen zum Netzwerk 113
Überlegungen zur Zeitplanung 114
Installation. 114
Einrichtung und Konfiguration 115
Einrichten eines UNIX-Agenten mit einem MPE-Master. 115
Einrichten eines MPE-FTA unter UNIX oder Windows NT. 115
Unterschiede bei der Verwendung verschiedener Plattformen 116
MPE-Master 117
Die Programme ″Arranger″ und ″Composer″ 117
Conman. 117
Fehlertolerante Agenten unter UNIX 117
Composer 117
Conman. 117
UNIX- oder Windows NT-Master. 118
Composer 118
Conman. 118
MPE-Slaves. 118

Arranger 118
Composer 118
Conman. 118

Anhang D. Auf Tivoli Workload Scheduler migrieren 119

Kurze Beschreibung von Tivoli Management Framework für neue Tivoli-Benutzer 119
Hinweise zur Installation 120
Installationsposition für den Client der Job Scheduling Console 120
Installationsposition für die Connector 120
Connector auf FTAs installieren. 121
Hinweise zum Sicherungs-Master. 121
Umgang mit nicht zu Ebene 1 gehörenden Mastern in einem vorhandenen Maestro-Netzwerk. 122
Den Sicherungs-Master versetzen. 122
Einen Sicherungs-Master erstellen 123
Datenbanken des Hauptdomänenmanagers anhängen. 123
Umgang mit nicht zu Ebene 1 gehörenden Sicherungs-Mastern in einem vorhandenen Maestro-Netzwerk. 124
Mehrere Fenster in der JS Console ausführen. 124
Eine vorhandene Sicherheitsdatei einsatzbereit machen. 124
Tivoli-Administratoren hinzufügen. 124
Sicherheit verwalten 127
Connector stoppen, um Änderungen zu implementieren 127
WMAEUTIL ausführen 127
Hinweise zu Zeitzonen. 128
Benutzervorgaben an andere Benutzer weitergeben (Angepasste Ansichten/Abfragen) 128
Erweiterte Datenbanken zurücksetzen. 130
Unter Windows NT migrieren 131

Glossar 133

Index 137

Vorwort

Das Handbuch *Tivoli Workload Scheduler Planung und Installation* enthält Informationen zur Installation eines Tivoli Workload Scheduler 8.1-Netzwerks. Dazu gehören Informationen zur Planung eines Netzwerks sowie zur Installation der Tivoli Workload Scheduler-Steuerkomponente, von TWS Connector und der grafischen Benutzerschnittstelle. Außerdem enthält es Anweisungen zum Anpassen der Tivoli Workload Scheduler-Optionen und der Sicherheitseinstellungen, um ein Netzwerk starten zu können. Darüber hinaus bietet es Tipps und Informationen zur Migration früherer Maestro-Versionen.

Zielgruppe

Dieses Handbuch richtet sich an folgende Zielgruppe:

- Tivoli Workload Scheduler-Administratoren - Benutzer, die die Struktur des Tivoli Workload Scheduler-Netzwerks planen
- Systemverantwortliche - Benutzer, die die verschiedenen Softwarepakete auf den Computern installieren, aus denen das Tivoli Workload Scheduler-Netzwerk besteht

Inhalt dieses Handbuchs

Das Handbuch *Tivoli Workload Scheduler Planung und Installation* besteht aus folgenden Kapiteln:

- Kapitel 1, Kapitel 1, „Einführung” auf Seite 1
  Enthält Planungsinformationen zur Einrichtung eines Tivoli Workload Scheduler-Netzwerks sowie eine Übersicht über den Installationsprozess.
- Kapitel 2, Kapitel 2, „Unter Windows NT installieren” auf Seite 19
  Erläutert die Installation der Tivoli Workload Scheduler-Steuerkomponente unter Windows NT.
- Kapitel 3, Kapitel 3, „Unter UNIX installieren” auf Seite 35
  Erläutert die Installation der Tivoli Workload Scheduler-Steuerkomponente unter UNIX.
- Kapitel 4, Kapitel 4, „Sicherheit definieren” auf Seite 49
  Erläutert die Anpassung der Sicherheitsdatei.
- Kapitel 5, Kapitel 5, „Optionale Anpassungsoptionen” auf Seite 69
  Erläutert die Anpassung Ihres Netzwerks und Ihrer Workstation nach der Installation.
- Anhang A, Anhang A, „Zeitzonen und Prüfung” auf Seite 89
  Erläutert die Konfiguration Ihrer Workstation für Zeitzonen und die Prüfung.
- Anhang B, Anhang B, „Tivoli Workload Scheduler-Netzwerke” auf Seite 99
  Erläutert die einem Tivoli Workload Scheduler-Netzwerk unterliegenden Mechanismen.
- Anhang C, Anhang C, „Netzwerkbetrieb mit MPE” auf Seite 113
  Erläutert die Konfiguration Ihres Netzwerks, wenn es MPE-Computer enthält.
- Anhang D, Anhang D, „Auf Tivoli Workload Scheduler migrieren” auf Seite 119
  Enthält hilfreiche Informationen zur Migration von Maestro.

Veröffentlichungen

In diesem Abschnitt werden Veröffentlichungen in der *Tivoli Workload Scheduler*-Bibliothek und andere zugehörige Dokumente aufgeführt. Er beschreibt auch, wie Sie online auf Tivoli-Veröffentlichungen zugreifen, wie Sie Tivoli-Veröffentlichungen bestellen und wie Sie Kommentare zu Tivoli-Veröffentlichungen abgeben.

Tivoli Workload Scheduler-Veröffentlichungen

Dieses Buch ist Teil einer umfangreichen Tivoli Workload Scheduler-Bibliothek. Diese Bücher können Sie bei der effektiven Verwendung von Tivoli Workload Scheduler unterstützen:

Tabelle 1. Liste der Tivoli Workload Scheduler-Veröffentlichungen

Task	Veröffentlichung	Bestellnummer
Tivoli Workload Scheduler planen und installieren	*Tivoli Workload Scheduler Planung und Installation*	SH12-3010-00
Die Tivoli Workload Scheduler-Befehlszeile verwenden, verstehen, wie erweiterte Agenten und Netzwerkagenten funktionieren, und Tivoli Workload Scheduler in NetView und Tivoli Business Systems Manager integrieren	*Tivoli Workload Scheduler Referenz*	SH12-3011-00
Tivoli Workload Scheduler-Fehlernachrichten verstehen	*Tivoli Workload Scheduler Error Messages*	SH19-4557
Fehlertolerante Tivoli Workload Scheduler-Agenten unter AS/400 installieren, konfigurieren und verwenden	*Tivoli Workload Scheduler AS/400 Benutzerhandbuch*	SH12-3012-00
Tivoli Workload Scheduler Plus Module einrichten und verwenden	*Tivoli Workload Scheduler Plus Module Benutzerhandbuch*	SH12-3013
Die neuesten Informationen zu Tivoli Workload Scheduler	*Tivoli Workload Scheduler Release-Informationen*	GI10-9726-00

Zugehörige Veröffentlichungen

Die folgenden Dokumente enthalten ebenfalls hilfreiche Informationen zu diesem Produkt, die sich auf die Endpunkt-zu-Endpunkt-Zeitplanung beziehen.

Tabelle 2. Liste der zugehörigen Veröffentlichungen

Task	Veröffentlichung	Bestellnummer
Die Tivoli Workload Scheduling-Produktgruppe verstehen	*Tivoli Workload Scheduler for z/OS General Information*	GH19-4539
Tivoli Workload Scheduler for z/OS planen	*Licensed Program Specifications*	GH19-4540
Kurzübersicht	*Tivoli Workload Scheduler for z/OS Quick Reference*	GH19-4541
Konzepte und Terminologie kennen lernen	*Tivoli Workload Scheduler for z/OS Getting Started*	SH19-4542
Tivoli Workload Scheduler für z/OS installieren	*Tivoli Workload Scheduler for z/OS Installation Guide*	SH19-4543
Die Auslastung planen und terminieren	*Tivoli Workload Scheduler for z/OS Planning and Scheduling the Workload*	SH19-4546
Tivoli Workload Scheduler für z/OS anpassen und optimieren	*Tivoli Workload Scheduler for z/OS Customization and Tuning*	SH19-4544

Tabelle 2. Liste der zugehörigen Veröffentlichungen (Forts.)

Task	Veröffentlichung	Bestellnummer
Anwendungsprogramme schreiben	*Tivoli Workload Scheduler for z/OS Programming Interfaces*	SH19-4545
Den aktuellen Plan steuern und überwachen	*Tivoli Workload Scheduler for z/OS Controlling and Monitoring the Workload*	SH19-4547
Nachrichten und Codes interpretieren	*Tivoli Workload Scheduler for z/OS Messages and Codes*	SH19-4548
Java-GUI - Letzte Änderungen	*Tivoli Job Scheduling Console Release-Informationen*	GI10-9725
Java-GUI verwenden	*Tivoli Workload Scheduler Console Benutzerhandbuch*	SH12-3009-00

Onlinebücher für Tivoli Workload Scheduler for z/OS

Alle Bücher der Bibliothek von Tivoli Workload Scheduler for z/OS, mit Ausnahme der lizenzierten Veröffentlichungen, sind als anzeigbare Softcopy auf CD-ROM im folgenden Softcopy-CKIT verfügbar:

- OS/390, SK2T-6951

Sie können die Softcopy-Bücher auf CD-ROMs lesen, indem Sie die folgenden IBM Lizenzprogramme verwenden:

- BookManager READ/2
- BookManager READ/DOS
- BookManager READ/6000

Alle BookManager-Programme benötigen einen Personal Computer mit einem CD-ROM-Laufwerk (das CD-ROMs lesen kann, die im ISO 9660-Standard formatiert sind) und einem passenden Adapter und Kabel. Zusätzliche Informationen zu Hardware und Software finden Sie in der Dokumentation des entsprechenden BookManager-Produkts, das Sie verwenden.

Aktualisierungen zu Büchern, die zwischen Releases veröffentlicht werden, stehen nur als Softcopy zur Verfügung.

Online auf Veröffentlichungen zugreifen

Sie können online auf viele Tivoli-Veröffentlichungen über die Tivoli-Kundenunterstützungs-Website unter folgender Adresse zugreifen:

http://www.tivoli.com/support/documents/

Diese Veröffentlichungen sind im PDF- und/oder HTML-Format verfügbar. Für einige Produkte sind auch übersetzte Dokumente verfügbar.

Veröffentlichungen bestellen

Sie können viele Tivoli-Veröffentlichungen online auf folgender Website bestellen:

http://www.ibm.com/shop/publications/order

Sie können die Veröffentlichungen auch telefonisch bestellen. Eine Liste der Telefonnummern für die verschiedenen Länder finden Sie auf folgender Website:

http://www.tivoli.com/inside/store/lit_order.html

Rückmeldung zu Veröffentlichungen

Wir interessieren uns auch für Ihre Erfahrungen mit den Tivoli-Produkten und -Handbüchern und sind Ihnen für Verbesserungsvorschläge dankbar. Wenn Sie Kommentare oder Vorschläge zu unseren Produkten und unserer Dokumentation haben, können Sie folgendermaßen mit uns in Kontakt treten:

- Senden Sie eine E-Mail an pubs@tivoli.com.
- Füllen Sie auf der folgenden Website unser Kundenrückmeldungsformular aus: http://www.tivoli.com/support/survey/

Plattformspezifische Informationen

In der folgenden Tabelle sind die Plattformversionen aufgeführt, die zum Zeitpunkt der Veröffentlichung des vorliegenden Dokuments unterstützt wurden. Ausführlichere und aktuelle Informationen hierzu finden Sie in den *Tivoli Workload Scheduler Release-Informationen*.

Plattform	Master	FTA	Connector	Job Scheduling Console
AIX 4.3.3s, 5.1	X	X	X	X
HP-UX 11.0, 11i	X	X	X	X
Solaris 2.7, 2.8	X	X	X	X
Microsoft Windows NT 4.0 >mit SP 5 oder SP 6a	X	X	X	X
Microsoft Windows Professional, Server, Advanced Server 2000 mit SP 1 oder SP 2		X		X
Microsoft Windows 98, Millennium				X
Compaq Tru64 5.1 (frühere OSF-Versionen)		X		
SGI Irix 6.5		X		
IBM Sequent Dynix 4.5		X		
Linux/INTEL Red Hat 7.1	X	X	X	X
Linux/390 SuSE 7.0		X		

Anmerkungen:

1. Eingeschränkte fehlertolerante Agenten werden auch auf der AS/400-Plattform unterstützt.
2. Die Liste der unterstützten Plattformen für die Job Scheduling Console ist auf die Unterstützung und die Verfügbarkeit von Java Runtime Environment (JRE) Version 1.3 beschränkt.

Kontaktaufname mit der Kundenunterstützung

Wenn Sie ein Problem mit einem Tivoli-Produkt haben, können Sie sich an die Tivoli-Kundenunterstützung wenden. Informationen dazu finden Sie im *Tivoli Customer Support Handbook* auf der folgenden Website:

http://www.tivoli.com/support/handbook/

In diesem Handbuch finden Sie Informationen dazu, wie Sie abhängig von der Schwere Ihres Problems die Kundenunterstützung kontaktieren. Außerdem enthält es folgende Informationen:

- Registrierung und Auswählbarkeit
- Telefonnummern und E-Mail-Adressen für Ihr Land
- Informationen, die Sie bereithalten sollten, wenn Sie Unterstützung anfordern

In diesem Handbuch verwendete Konventionen

In diesem Handbuch werden verschiedene Schriftbildkonventionen für besondere Begriffe und Aktionen verwendet. Diese Konventionen sind wie folgt festgelegt:

Informationsart	Konvention	Beispiel
Befehle	Nur Großbuchstaben	CREATE
Verweise im Text auf Felder in Anzeigen	Nur Großbuchstaben	MENGE
Eingaben, die Sie in den Feldern von Anzeigen vornehmen müssen	Monospace-Schrift	`MEINEANWENDUNG`
Erstes Auftreten eines neu eingeführten Begriffs	Kursiv	*Anwendung*

1

Einführung

Dieses Handbuch enthält Anweisungen zur Installation und Aktualisierung von Tivoli Workload Scheduler Version 8.1.

Neue Funktionen in dieser Version

Tivoli Workload Scheduler Version 8.1 umfasst die folgenden funktionalen Erweiterungen:

- Mehrere Kalender mit Feiertagen
- Regel für arbeitsfreie Tage
- Leistungsoptimierungen
- Installationsoptimierungen
- Linux-Unterstützung
- Endpunkt-zu-Endpunkt-Zeitplanung
- Integration in Tivoli Business Systems Manager

Mehrere Kalender mit Feiertagen

Der Kalender mit arbeitsfreien Tagen, wobei ein arbeitsfreier Tag das Gegenteil von einem Arbeitstag ist, erweitert jetzt die Funktion des Kalenders **Holidays**, da Sie mit seiner Hilfe die Bedeutung von Arbeitstagen innerhalb von Tivoli Workload Scheduler anpassen können. In früheren Versionen war die Definition der Arbeitstage durch den Kalender **Holidays** festgelegt, sofern dieser vorhanden war.

Mit dieser neuen Funktion können Sie auswählen, ob ein Jobnetz oder ein Befehl die Standarddefinition der Arbeitstage auf der Grundlage des Kalenders **Holidays** oder eine angepasste Definition der Arbeitstage, die von einem bestimmten Kalender mit arbeitsfreien Tagen angegeben wird, verwenden soll. Wenn Sie für ein bestimmtes Jobnetz oder einen bestimmten Befehl Ihren eigenen Kalender mit arbeitsfreien Tagen verwenden wollen, ist die neue Definition der Arbeitstage, wie sie durch den Kalender mit arbeitsfreien Tagen angegeben wird, nur auf das bestimmte Jobnetz bzw. den bestimmten Befehl beschränkt. Wenn Sie für das Jobnetz oder den Befehl keinen Kalender mit arbeitsfreien Tagen angeben, dann wird der Kalender **Holidays** verwendet und die Arbeitstage behalten ihre ursprüngliche Bedeutung.

Standardmäßig wird das Schlüsselwort **workdays** wie folgt definiert:

```
workdays bedeutet täglich mit Ausnahme von Samstag, Sonntag und allen Daten, die im Kalender
HOLIDAYS angegeben sind.
```

Wenn Sie einen Kalender mit arbeitsfreien Tagen in einer Jobnetzdefinition angeben, basiert das Schlüsselwort **workdays** für dieses Jobnetz auf diesem Kalender anstatt auf dem Kalender **Holidays**. Jedes Auftreten des Schlüsselworts **workdays** in diesem Jobnetz wird wie folgt interpretiert:

```
workdays bedeutet täglich mit Ausnahme von Samstag, Sonntag und allen Daten, die im
entsprechenden Kalender mit arbeitsfreien Tagen angegeben sind.
```

Standardmäßig werden Samstag und Sonntag nicht als Arbeitstage betrachtet, Sie können allerdings angeben, ob sie als Arbeitstage gelten sollen.

Wenn Sie in der Jobnetzdefinition keinen Kalender mit arbeitsfreien Tagen angeben, wird die Standarddefinition für das Schlüsselwort **workdays** verwendet. Diese lautet wie folgt:

```
workdays bedeutet täglich mit Ausnahme von Samstag, Sonntag und allen Daten, die im Kalender
HOLIDAYS angegeben sind.
```

Sie können Kalender mit arbeitsfreien Tagen über die Tivoli Workload Scheduler-Befehlszeilenschnittstelle (CLI) oder über die Job Scheduling Console definieren.

Regel für arbeitsfreie Tage

Sie können eine Regel für arbeitsfreie Tage angeben, wenn Sie einen Laufzyklus für ein Jobnetz definieren. Die Regel legt fest, was der Scheduler zu tun hat, wenn das Plandatum eines Jobnetzes auf einen arbeitsfreien Tag fällt. Wenn das Plandatum auf einen arbeitsfreien Tag fällt, bestehen folgende Möglichkeiten für Tivoli Workload Scheduler:

- Das Jobnetz ausführen
- Das Jobnetz nicht ausführen
- Das Jobnetz am letzten Arbeitstag vor dem arbeitsfreien Tag ausführen
- Das Jobnetz am ersten Arbeitstag nach dem arbeitsfreien Tag ausführen

Die Regel für arbeitsfreie Tage ist eng mit der Verwendung mehrerer ON- und EXCEPT-Klauseln in der Planungssprache verbunden. Praktisch gesehen bedeutet die Verwendung der ON- oder EXCEPT-Klauseln innerhalb einer Jobnetzdefinition (bzw. einer Plandefinition), dass ein Laufzyklus dafür definiert wird.

Wenn Sie keine Regel für arbeitsfreie Tage angeben, verwendet Tivoli Workload Scheduler seinen Standardmechanismus und führt das Jobnetz aus, auch wenn das ausgewählte Ausführungsdatum auf einen arbeitsfreien Tag fällt.

Tivoli Workload Scheduler plant dasselbe Jobnetz nicht mehr als einmal an einem bestimmten Produktionstag erneut, falls sein Bearbeitungsdatum wegen der Anwendung einer Regel für arbeitsfreie Tage verschoben wurde.

Weitere Informationen finden Sie in den Handbüchern *Tivoli Workload Scheduler Referenz* oder *Tivoli Workload Scheduler Console Benutzerhandbuch*.

Leistungsoptimierungen

Die neuen Leistungserweiterungen kommen besonders in Tivoli Workload Scheduler-Netzwerken mit vielen Workstations, umfangreichen Zeitplänen und komplexen Beziehungen zwischen Zeitplanungsobjekten zur Geltung. Die Optimierungen betreffen die folgenden Bereiche:

- Tagesplanerstellung: **Jnextday** wird schneller ausgeführt und folglich kann der Hauptdomänenmanager seine Produktionstasks schneller starten.
- Tagesplanverteilung: Der Tivoli Workload Scheduler-Administrator kann jetzt die Komprimierung der Datei **Symphony** aktivieren, so dass der Tagesplan früher auf andere Knoten verteilt werden kann.
- E/A-Optimierung: Tivoli Workload Scheduler führt weniger Dateizugriffe aus und optimiert die Verwendung von Systemressourcen. Die Optimierungen beziehen sich auf folgende Punkte:

- Ereignisdateien: Die Antwortzeit bei Ereignissen wurde verbessert, so dass der Nachrichtenfluss schneller ist.
- Tagesplan: Der Lese- und Schreibzugriff auf die Datei **Symphony** ist schneller. Der Tagesplan kann daher schneller, als dies früher der Fall war, aktualisiert werden.

Informationen zur Implementierung dieser funktionalen Erweiterungen finden Sie unter „Lokale Optionen” auf Seite 74.

Installationsoptimierungen

Die Installation von Netman ist unter Windows NT nicht länger ein separater Prozess. Sie ist jetzt Teil der Installationsschritte des Produkts.

Linux-Unterstützung

Version 8.1 von Tivoli Workload Scheduler fügt die Unterstützung für folgende Linux-Plattformen hinzu:

- Linux for Intel als Hauptdomänenmanager und fehlertoleranter Agent
- Linux for S/390 als fehlertoleranter Agent

Endpunkt-zu-Endpunkt-Zeitplanung

Die Endpunkt-zu-Endpunkt-Zeitplanung ermöglicht Ihnen, Jobs in Großrechner-, Windows- und UNIX-Umgebungen für echte verteilte Zeitplanung zu planen und zu steuern. In der Endpunkt-zu-Endpunkt-Konfiguration wird Tivoli Workload Scheduler for z/OS als Scheduler für die Jobplanungsumgebung verwendet. Die Domänenmanager und fehlertoleranten Agenten von Tivoli Workload Scheduler werden zum Terminieren auf den verteilten Plattformen verwendet.

Die Endpunkt-zu-Endpunkt-Zeitplanung basiert auf der Möglichkeit, einen Tivoli Workload Scheduler-Domänenmanager und seine zugehörigen Agenten und Domänen mit der Tivoli Workload Scheduler for z/OS-Steuerkomponente zu verbinden. Die Tivoli Workload Scheduler for z/OS-Steuerkomponente erstellt den Produktionsplan auch für das verteilte Netzwerk und sendet ihn an den Domänenmanager. Der Domänenmanager sendet eine Kopie des Plans zur Ausführung an jeden seiner verbundenen Agenten und untergeordneten Domänenmanager. Der Domänenmanager fungiert als das Brokersystem für das gesamte verteilte Netzwerk, indem er alle Abhängigkeiten des Netzwerks auflöst. Er sendet seine Aktualisierungen (in Form von Ereignissen) an Tivoli Workload Scheduler for z/OS, so dass der Plan entsprechend aktualisiert werden kann. Tivoli Workload Scheduler for z/OS handhabt seine eigenen Jobs und benachrichtigt den Domänenmanager über alle Statusänderungen der Jobs von Tivoli Workload Scheduler for z/OS, die den Tivoli Workload Scheduler-Plan betreffen. In dieser Konfiguration erkennen der Domänenmanager und alle verteilten Agenten Tivoli Workload Scheduler for z/OS als den Hauptdomänenmanager an und benachrichtigen ihn über alle Änderungen in ihren eigenen Plänen. Gleichzeitig sind die Agenten nicht berechtigt, die Jobs von Tivoli Workload Scheduler for z/OS zu beeinflussen, da diese Jobs als auf dem Master ausgeführt betrachtet werden und der Master der einzige Knoten ist, der für sie zuständig ist.

Die Implementierung der Endpunkt-zu-Endpunkt-Zeitplanung wird primär auf Tivoli Workload Scheduler for z/OS durchgeführt. Für die fehlertoleranten Agenten von Tivoli Workload Scheduler müssen lediglich die unter Kapitel 2, „Unter Windows NT installieren” auf Seite 19 und Kapitel 3, „Unter UNIX installieren” auf Seite 35 beschriebenen Optionen gesetzt werden.

Integration in Tivoli Business Systems Manager

Tivoli Business Systems Manager ist eine objektorientierte Systemmanagementanwendung, die die Überwachung und das Ereignismanagement von Ressourcen, Anwendungen und Subsystemen innerhalb des Unternehmens mit dem Ziel, fortlaufende Verfügbarkeit bereitzustellen, bietet. Durch die Überwachung der Tivoli Workload Scheduler-Tagespläne mit Tivoli Business Systems Manager können Probleme rasch festgestellt werden, die die erfolgreiche und rechtzeitige Beendigung der Pläne gefährden können. Die Integration von Tivoli Workload Scheduler in Tivoli Business Systems Manager ermöglicht es, die Pläne aus einer einheitlichen Geschäftssystemperspektive zu verwalten.

Die Integration wird durch Folgendes erreicht:

- Eine spezielle Markierung (Schlüsselmarkierung) in Tivoli Workload Scheduler, die die Jobs und Jobnetze markiert, die Tivoli Business Systems Manager genauer überwachen soll. Informationen zu diesen Schlüsselobjekten wird in Form von Ereignissen gesendet.
- Einen Mechanismus für Massenerfassung, der Informationen zu allen wichtigen Jobs und Jobnetzen des Tagesplans an Tivoli Business Systems Manager sendet.
- Einen Mechanismus für Deltaerfassung, der die Tagesplanänderungen für wichtige Jobs und Jobnetze an Tivoli Business Systems Manager sendet.

Näheres dazu finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

Gelöschte Funktionalitäten

Folgende Funktionalitäten sind nicht mehr verfügbar:

- Die traditionellen Maestro-GUIs, einschließlich GCONMAN und GCOMPOSER
- Visual Maestro
- Remote Console
- Unterstützung für HP OpenView
- IT Operation (ehemals HP OpC)
- Tivoli GEM Instrumentation

Tivoli Workload Scheduler

Tivoli Workload Scheduler besteht aus drei Komponenten:

Tivoli Workload Scheduler-Steuerkomponente
Die Steuerkomponente wird auf verschiedenen Microsoft Windows- und UNIX-Betriebssystemen ausgeführt. Installieren Sie die Steuerkomponente auf dem Master und auf den CPUs aller physischen Agenten.

Tivoli Workload Scheduler Connector
Ordnet der Tivoli Workload Scheduler-Steuerkomponente Befehle der Job Scheduling Console zu. Installieren Sie den Connector auf dem Master und auf den fehlertoleranten Agenten, die Sie als Sicherungsmaschinen für die Master-CPU verwenden. Für die Verwendung des Connectors wird Tivoli Management Framework benötigt, das für einen Tivoli-Server oder einen verwalteten Tivoli-Knoten konfiguriert ist.

Job Scheduling Console (JS Console)
Eine grafische Benutzerschnittstelle für Tivoli Workload Scheduler, die auf Java basiert. Installieren Sie die Job Scheduling Console auf Maschinen, von denen aus Sie Plan- und Datenbankobjekte verwalten möchten. Für die Verwendung der Job

Scheduling Console müssen die Tivoli Workload Scheduler-Steuerkomponente oder der Tivoli Workload Scheduler Connector nicht notwendigerweise auf derselben Maschine installiert sein. Sie können die Job Scheduling Console von jeder Maschine aus verwenden, wenn diese über TCP/IP mit der Maschine verbunden ist, auf der der Tivoli Workload Scheduler-Connector ausgeführt wird.

Ein Tivoli Workload Scheduler-Netzwerk besteht aus den Workstations oder CPUs, auf denen Jobs und Jobnetze ausgeführt werden.

Workstationdefinitionen verweisen in erster Linie auf physische Workstations. Bei den Workstations von erweiterten Agenten und Netzwerkagenten dagegen handelt es sich um logische Definitionen, die einer physischen Tivoli Workload Scheduler-Workstation zugeordnet sein müssen.

Ein Tivoli Workload Scheduler-Netzwerk besteht aus folgenden Workstationtypen:

Hauptdomänenmanager (Master Domain Manager, MDM)
Der Domänenmanager der obersten Domäne eines Tivoli Workload Scheduler-Netzwerks. Er enthält die zentralen Datenbankdateien, die zur Dokumentierung von Zeitplanungsobjekten verwendet werden. Darüber hinaus erstellt er den Produktionsplan bei jedem Tagesstart und führt sämtliche Protokollierungs- und Berichtsfunktionen für das Netzwerk aus.

Sicherungs-Master
Ein fehlertoleranter Agent, der die Rolle des Hauptdomänenmanagers übernehmen kann.

Domänenmanager
Der Management-Hub in einer Domäne. Die gesamte Kommunikation von Agenten in der Domäne wird über den Domänenmanager geleitet.

Sicherungsdomänenmanager
Ein fehlertoleranter Agent, der die Rolle seines Domänenmanagers übernehmen kann.

Fehlertoleranter Agent (FTA)
Eine Workstation, die lokale Abhängigkeiten auflösen und zugehörige Jobs ohne Domänenmanager starten kann.

Standardagent
Eine Workstation, die Jobs nur unter der Steuerung des zugehörigen Domänenmanagers startet.

Erweiterter Agent
Eine logische Workstationdefinition, mit deren Hilfe Jobs auf anderen Systemen und in anderen Anwendungen (z. B. Baan, Peoplesoft, Oracle, SAP sowie JES2 und JES3 for z/OS) gestartet und gesteuert werden können.

Netzwerkagent
Eine logische Workstationdefinition für die Erstellung von Abhängigkeiten zwischen Jobs und Jobnetzen in verschiedenen Tivoli Workload Scheduler-Netzwerken.

Job Scheduling Console-Client
Eine Workstation mit der grafischen Benutzerschnittstelle, mit deren Hilfe Planer und Operatoren Plan- und Datenbankobjekte verwalten.

In der nachfolgenden Tabelle wird zusammengefasst, welche Komponente auf welcher Workstation installiert wird:

Workstation	Steuer-komponente	Connector	Job Scheduling Console
Hauptdomänenmanager	Ja	Ja	Optional
Sicherungs-Master	Ja	Ja	Optional
Domänenmanager	Ja	Optional	Optional
Sicherungsdomänenmanager	Ja	Optional	Optional
Fehlertoleranter Agent	Ja	Optional	Optional
Standardagent	Ja	Nein	Optional
Erweiterter Agent	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
Netzwerkagent	Nicht zutreffend	Nicht zutreffend	Nicht zutreffend
Job Scheduling Console-Client	Nein	Nein	Ja

Job Scheduling Console

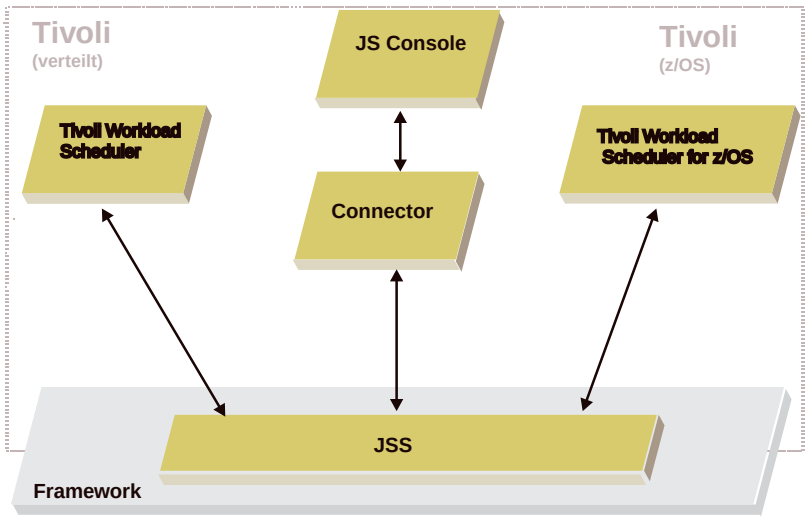
Bei der Job Scheduling Console handelt es sich um eine grafische Benutzerschnittstelle, über die Plan- und Datenbankobjekte verwaltet werden können. In dieser Schnittstelle wird über Tivoli Workload Scheduler Connector die Funktionalität von Conman und Composer zur Verfügung gestellt. Sie kann allein oder zusammen mit den Befehlszeilenschnittstellen (Command Line Interfaces, CLI) ausgeführt werden.

Während die Verwendung von CLI-Befehlen auf Tivoli Workload Scheduler-Workstations beschränkt ist, kann die Job Scheduling Console auf Maschinen außerhalb des Tivoli Workload Scheduler-Netzwerks installiert werden. Zur Ausführung der Job Scheduling Console muss nur die Möglichkeit bestehen, sich an einer Tivoli Workload Scheduler-Workstation anzumelden, auf der Connector ausgeführt wird. Das bedeutet, dass Sie Plan- und Datenbankobjekte nicht nur vom Master oder den Agentenworkstations aus verwalten können, sondern von beliebigen Systemen aus, einschließlich Laptops, auf denen die Job Scheduling Console installiert ist und von denen aus Sie über TCP/IP eine Verbindung zum Master oder zum fehlertoleranten Agenten, auf dem der Connector ausgeführt wird, herstellen können.

Von derselben Job Scheduling Console können Sie auch Plan- und Datenbankobjekte von Tivoli Workload Scheduler for z/OS verwalten, vorausgesetzt, die Anmeldung an einer Maschine, auf der der Tivoli Workload Scheduler for z/OS-Connector ausgeführt wird, ist möglich.

Eine ausführliche Beschreibung der Installation der Job Scheduling Console finden Sie im Handbuch *Tivoli Workload Scheduler Console Benutzerhandbuch*.

In der folgenden Abbildung wird die Beziehung zwischen Tivoli Management Framework, den Job Scheduling Services, den Tivoli Workload Scheduler- und Tivoli Workload Scheduler for z/OS-Job Scheduling-Steuerkomponenten und der Job Scheduling Console dargestellt.



Die Job Scheduling Console verbessert die Integration zwischen den beiden Zeitplanungs-Steuerkomponenten, da sie einen einzelnen Zugriffspunkt für Datenbank- und Planobjekte auf dem Großrechner und den verteilten Plattformen bietet. Dies ist vor allem dann hilfreich, wenn Sie die Funktion der Endpunkt-zu-Endpunkt-Zeitplanung von Tivoli Workload Scheduler for z/OS verwenden.

Terminologie der Job Scheduling Console

Die Terminologie der Job Scheduling Console unterscheidet sich von der für die Befehlszeile und der in den traditionellen Maestro-Versionen verwendeten Terminologie. In der nachfolgenden Tabelle finden Sie die alten Begriffe und die entsprechenden Begriffe für die Job Scheduling Console. Weitere Definitionen finden Sie im Glossar.

Befehlzeile	Job Scheduling Console	Definition
Zeitplan	Jobnetz	Eine Arbeitseinheit, die aus einer Reihe von Jobs und deren Abhängigkeiten besteht
	Jobnetz-Instanz	Das Vorkommen eines Jobnetzes im Plan
Job	Job	Eine ausführbare Datei, eine Task bzw. ein Befehl und die zugehörigen Attribute. Der Job ist für die Ausführung als Teil eines Jobnetzes terminiert.
	Job-Instanz	Das Vorkommen eines Jobs im Plan
CPU	Workstation	Ein logischer Prozessor (in der Regel ein Computer), der Jobs ausführt. Es gibt verschiedene Workstationtypen: Domänenmanager, Sicherungsdomänenmanager, fehlertolerante Agenten, Standardagenten und erweiterte Agenten.
Mozart-Dateien	Datenbank	Eine Sammlung von Zeitplanungsobjekten, die Jobs, Jobnetze, Parameter, Kalender und Ressourcen enthält. Der Zugriff erfolgt über *Composer*, früher *Gcomposer*.

Befehlzeile	Job Scheduling Console	Definition
Symphony-Datei	Plan	Die für einen bestimmten Zeitraum (in der Regel 24 Stunden) terminierten Aktivitäten. Der Plan wird fortlaufend aktualisiert und gibt daher immer den aktuellen Status aller Aktivitäten in Tivoli Workload Scheduler zurück. Der Zugriff erfolgt über *Console Manager*, früher *Gconman*.
AT-Zeit	Startzeit	Der Zeitpunkt, zu dem die Ausführung eines Jobs bzw. Jobnetzes frühestens beginnt
UNTIL-Zeit	Termin	Der Zeitpunkt, zu dem die Ausführung eines Jobs bzw. Jobnetzes spätestens beginnt
ON- und EXCEPT-Daten	Laufzyklen	Die Zeitpunkte, zu denen ein Jobnetz ausgeführt bzw. von der Ausführung ausgeschlossen wird

Überlegungen zu Zeitzonen

Die Zeitzonenunterstützung kann wahlweise aktiviert oder inaktiviert werden. Die Zeitzonenfunktion ist aktiviert, wenn die Markierung *timezone enable* in der Datei 'globalopts' auf **Ja** gesetzt ist und bei der Master-Workstation eine Zeitzone definiert ist.

Standardmäßig ist die Zeitzonenfunktion bei der Installation inaktiviert.

Ist die Zeitzonenfunktion aktiviert, sind auch die entsprechenden Felder in der Job Scheduling Console aktiviert. Ist die Zeitzonendefinition einer Workstation leer, wird für diese automatisch der Wert der Workstation des Hauptdomänenmanagers übernommen.

Ist die Zeitzonenfunktion inaktiviert, sind in der gesamten Anwendung alle Zeitzonenfelder ebenfalls inaktiviert. In diesem Fall erhalten Sie bei dem Versuch, eine Zeitzone plus Zeitangabe in Composer und Console Manager einzugeben, eine Fehlermeldung. In der Job Scheduling Console sind die Zeitzonenfelder sowohl für die Datenbank als auch für den Plan inaktiviert. Die einzige Ausnahme sind Workstations in der Datenbank, bei denen die Zeitzonenfunktion immer aktiviert ist.

Bei der Aktivierung der Zeitzone für die Master-Workstation wird empfohlen, dass Sie für fehlertolerante Agenten, die sich in einer anderen Zeitzone als die Master-Workstation befinden, eine Zeitzone angeben. Erfolgt keine Zeitzonenangabe für einen fehlertoleranten Agenten, wird davon ausgegangen, dass für den Agenten dieselbe Zeitzone wie für die Master-Workstation gilt.

Nach der Aktivierung der Zeitzonen können Sie eine Zeitzone und eine Zeit in Console Manager, Composer oder in der Job Scheduling Console angeben. Bei Angabe einer Zeit ohne Zeitzone wird letztere von der Workstation übernommen, auf der der Job bzw. das Jobnetz ausgeführt wird; ist auf dieser keine Zeitzone angegeben, wird die der Master-Workstation übernommen.

Netzwerkplanung

Klären Sie die nachfolgenden Fragen, bevor Sie mit der Installation von Tivoli Workload Scheduler beginnen. In den nächsten Abschnitten finden Sie Informationen, die Ihnen bei der Klärung dieser Fragen entsprechend Ihrer Bedürfnisse behilflich sind.

1. Verwenden Sie eine Netzwerkstruktur mit einer oder mehreren Domänen?
2. Wenn Sie mehrere Domänen verwenden, wie möchten Sie diese einteilen?
   - Nach geografischen Standorten, z. B. Domänen in London und Paris
   - Nach Zeitzonen, z. B. PST (Pacific Standard Time) und EST (Eastern Standard Time)
   - Nach Geschäftsbereichen, z. B. Marketing und Buchhaltung
3. Verwenden Sie erweiterte oder nicht erweiterte Datenbanken?
4. Verwenden Sie die Zeitzonenfunktion?

Übersicht über ein Tivoli Workload Scheduler-Netzwerk

Ein Workload Scheduler-Netzwerk besteht aus mindestens einer Workload Scheduler-Domäne, der Hauptdomäne, in der der Hauptdomänenmanager der Management-Hub ist. Um ein weit verteiltes Netzwerk in kleinere, lokal verwaltete Gruppen zu unterteilen, können zusätzliche Domänen verwendet werden.

Durch Verwendung mehrerer Domänen nimmt der Datenverkehr auf dem Netzwerk ab, da die Kommunikation zwischen dem Hauptdomänenmanager und anderen Computern reduziert wird.

In einer Konfiguration mit nur einer Domäne kommuniziert der Hauptdomänenmanager mit allen Workstations im Workload Scheduler-Netzwerk.

In einer Konfiguration mit mehreren Domänen kommuniziert der Hauptdomänenmanager jedoch nur mit den Workstations innerhalb seiner Domäne und mit untergeordneten Domänenmanagern. Die untergeordneten Domänenmanager kommunizieren ihrerseits mit Workstations innerhalb ihrer Domänen und untergeordneten Domänenmanagern. Durch den Einsatz mehrerer Domänen wird außerdem die Fehlertoleranz erhöht, da sich die Probleme durch den Verlust eines Domänenmanagers auf eine Domäne beschränken. Um diese Gefahr weiter zu minimieren, können Sie Sicherungsdomänenmanager zuweisen, die bei Ausfall des Domänenmanagers dessen Funktionen übernehmen.

Funktionalität der Domänen

Bei der Definition einer neuen Domäne müssen Sie die übergeordnete Domäne und den Domänenmanager angeben. Bei der übergeordneten Domäne handelt es sich um die Domäne, die in der Domänenhierarchie direkt über der neuen Domäne steht. Die gesamte Domänenkommunikation wird über den übergeordneten Domänenmanager geleitet.

Der Hauptdomänenmanager erstellt vor Beginn eines Produktionstages eine Produktionssteuerungsdatei, die sogenannte Symphony-Datei. Anschließend wird innerhalb des Netzwerks ein Neustart von Tivoli Workload Scheduler durchgeführt, und der Hauptdomänenmanager sendet eine Kopie der neuen Symphony-Datei an jeden Agenten, der automatisch mit ihm verbunden ist, sowie an untergeordnete Domänenmanager. Die Domänenmanager senden ihrerseits Kopien an die Agenten, die automatisch mit ihnen verbunden sind, sowie an untergeordnete Domänenmanager.

Wenn das Netzwerk gestartet wurde, werden Planungsnachrichten, wie z. B. Nachrichten zum Jobbeginn und -ende, von den Agenten an die jeweiligen Domänenmanager übergeben und anschließend über die übergeordneten Domänenmanager an den Hauptdomänenmanager übermittelt. Der Hauptdomänenmanager verteilt diese Nachrichten dann über die hierarchische Baumstruktur, um die Symphony-Dateien von Domänenmanagern und fehlertoleranten Agenten im Modus 'Vollständiger Status' zu aktualisieren.

Auf eine Domäne begrenzte Verarbeitung

Bei der Planung der Tivoli Workload Scheduler-Domänen ist das Konzept der begrenzten Verarbeitung von entscheidender Bedeutung. Ziel dabei ist die Trennung bzw. Begrenzung der Planungsaktivitäten, basierend auf bestimmten Gemeinsamkeiten.

Bei diesen Gemeinsamkeiten kann es sich z. B. um geografische Standorte, Geschäftsfunktionen und Anwendungsgruppierungen handeln. Durch eine solche gruppierte Verarbeitung kann das Informationsvolumen, das zwischen den Domänen ausgetauscht werden muss, reduziert werden. Aus der Verarbeitungsbegrenzung in Domänen ergeben sich folgende Vorteile:

- Verringerter Datenaustausch auf dem Netzwerk. Da die Verarbeitung auf die Domäne beschränkt wird, ist die Kommunikation zwischen den Domänen seltener erforderlich.
- Verbesserung der Sicherheit und Vereinfachung der Verwaltung. Die Sicherheits- und Verwaltungsaufgaben können auf der Domänenebene definiert und auf sie beschränkt werden. An Stelle einer netzwerkweiten oder einer workstationspezifischen Verwaltung können Sie sich für die Domänenverwaltung entscheiden.
- Die Fehlertoleranz im Netzwerk und auf den Workstations kann optimiert werden. In einem TWS-Netzwerk mit mehreren Domänen können Sie einen Sicherungsdomänenmanager für jeden Domänenmanager definieren, wodurch Probleme auf eine Domäne beschränkt werden und die Verarbeitung in anderen Domänen nicht beeinträchtigen.

Überlegungen zum Netzwerk

Die folgenden Fragen sollen Ihnen bei der Einrichtung Ihres Tivoli Workload Scheduler-Netzwerks helfen. Manche Fragen beziehen sich auf das Netzwerk, andere auf die Anwendungen, die von TWS gesteuert werden. Möglicherweise müssen Sie sich zur Klärung bestimmter Punkte mit anderen Personen in Ihrem Unternehmen absprechen.

- Wie groß ist Ihr TWS-Netzwerk? Aus wie vielen Computern besteht es? Wie viele Anwendungen und Jobs werden ausgeführt?

  Die Größe Ihres Netzwerks ist bei der Entscheidung, ob Sie eine Architektur mit einer Domäne oder mit mehreren Domänen verwenden, von Bedeutung. Wenn Sie über relativ wenige Computer bzw. über relativ wenige Anwendungen verfügen, die von TWS gesteuert werden, ist ein Netzwerk mit mehreren Domänen möglicherweise nicht erforderlich.

- Wie viele geografische Standorte müssen von Ihrem TWS-Netzwerk abgedeckt werden? Wie zuverlässig und effizient ist die Kommunikation zwischen den Standorten?

  Hierbei handelt es sich um einen der wichtigsten Gründe für eine Architektur mit mehreren Domänen. Für gewöhnlich wird jedem geografischen Standort eine Domäne zugeordnet. Wenn Sie sich für eine Architektur mit einer Domäne entscheiden, sind Sie zur Gewährleistung einer kontinuierlichen Verarbeitung in einem höheren Maß vom Netzwerk abhängig.

- Benötigen Sie eine zentrale oder dezentrale Verwaltung von Tivoli Workload Scheduler?

  In einem Tivoli Workload Scheduler-Netzwerk (unabhängig davon, ob es sich um ein Netzwerk mit einer Domäne oder mit mehreren Domänen handelt) können Sie TWS von einem einzigen Knoten, dem Hauptdomänenmanager, aus verwalten. Wenn Sie mehrere Standorte unabhängig voneinander verwalten möchten, sollten Sie möglicherweise an jedem Standort ein eigenständiges TWS-Netzwerk installieren. Bis zu einem gewissen Grad ist eine dezentrale Verwaltung in einem eigenständigen TWS-Netzwerk möglich, da Dateisysteme angehängt oder gemeinsam verwendet werden können.

- Ist der Standort in mehrere physische oder logische Definitionseinheiten unterteilt? Gibt es verschiedene Gebäude, und bestehen diese aus mehreren Stockwerken? Gibt es verschiedene Abteilungen bzw. Geschäftsfunktionen? Gibt es verschiedene Anwendungen?

  All dies können Gründe für eine Konfiguration mit mehreren Domänen sein. Es könnte z. B. eine Domäne für jedes Gebäude, für jede Abteilung, Geschäftsfunktion oder Anwendung geben (Produktion, Buchhaltung, Entwicklung usw.).

- Verfügen Sie über Anwendungen wie z. B. SAP R/3 oder Baan, die mit Tivoli Workload Scheduler ausgeführt werden sollen?

  Wenn es sich dabei um Einzelanwendungen handelt, die unabhängig von anderen Anwendungen ausgeführt werden, ist es unter Umständen sinnvoll, wenn sie sich in einer eigenständigen TWS-Domäne befinden.

- Sollen Ihre TWS-Domänen Ihre Windows NT-Domänen widerspiegeln?

  Dies ist zwar nicht unbedingt erforderlich, könnte aber nützlich sein.

- Möchten Sie bestimmte Systemgruppen auf Grund von leistungsbezogenen oder anderen Kriterien isolieren?

  Dieser Punkt spricht ebenfalls für ein TWS-Netzwerk mit mehreren Domänen, da so Systeme je nach Leistung oder Plattform zusammengefasst werden können.

- Wie hoch ist der aktuelle Datenaustausch auf dem Netzwerk?

  Ist die Verwaltung des Datenaustauschs auf dem Netzwerk ohne Probleme möglich, sinkt die Notwendigkeit mehrerer Domänen.

- Gehen Ihre Jobabhängigkeiten über System- und Anwendungsgrenzen sowie über geografische Grenzen hinaus? Hängt z. B. der Start von Job1 auf CPU3 vom Abschluss von Job2 auf CPU4 ab?

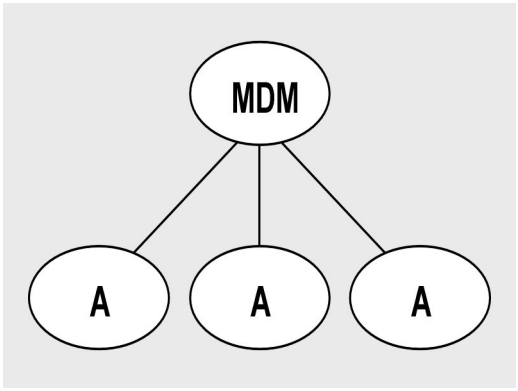
  Der Grad an gegenseitiger Abhängigkeit zwischen verschiedenen Jobs ist bei der Planung eines TWS-Netzwerks von erheblicher Bedeutung. Wenn Sie ein Netzwerk mit mehreren Domänen verwenden, sollten sich voneinander abhängige Objekte innerhalb derselben Domäne befinden. Somit nimmt der Datenaustausch auf dem Netzwerk ab, und die Domänenarchitektur kann besser genutzt werden.

- Welcher Fehlertoleranzgrad wird benötigt?

  Ein offensichtlicher Nachteil einer Konfiguration mit nur einer Domäne ist die Abhängigkeit von nur einem Domänenmanager. In einem Netzwerk mit mehreren Domänen betrifft der Ausfall eines Domänenmanagers nur die Agenten innerhalb seiner Domäne.

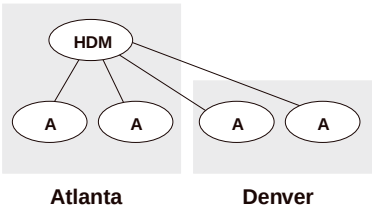
Netzwerke mit einer Domäne

Ein TWS-Netzwerk mit einer Domäne besteht aus einem Hauptdomänenmanager und einer beliebigen Anzahl an Agenten. Das unten stehende Diagramm stellt ein Beispiel für ein Netzwerk mit einer Domäne dar. Ein solches Netzwerk eignet sich für Unternehmen mit einer begrenzten Anzahl an Standorten und Geschäftsfunktionen. Die gesamte Kommunikation innerhalb des Netzwerkes wird über den Hauptdomänenmanager geleitet. Bei lediglich einem Standort ist nur die Zuverlässigkeit Ihres lokalen Netzwerkes und die Menge an Daten, die dieses verarbeiten kann, von Bedeutung.

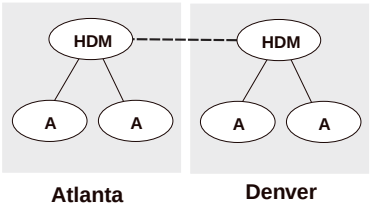


Bitte beachten Sie, dass Netzwerke mit einer Domäne problemlos mit anderen Netzwerken mit einer oder mehreren Domäne(n) kombiniert werden können, um den Anforderungen von Strukturen mit mehreren Standorten zu entsprechen.

Tivoli Workload Scheduler unterstützt INET-Abhängigkeiten von Jobs, die in verschiedenen TWS-Netzwerken ausgeführt werden.



Oder:



Das erste Beispiel stellt ein Netzwerk mit einer Domäne dar. Der Standort des Hauptdomänenmanagers ist Atlanta, hier befinden sich auch mehrere Agenten. Auch in Denver gibt es Agenten. Die Agenten in Denver hängen vom Hauptdomänenmanager in Atlanta ab, der alle Abhängigkeiten zwischen den Agenten auflösen muss, auch wenn sich diese Abhängigkeiten bei Jobs ergeben, die in Denver ausgeführt werden. Eine mögliche Alternative wäre die Einrichtung getrennter Tivoli Workload Scheduler-Netzwerke mit jeweils einer Domäne in Atlanta und Denver, wie im zweiten Beispiel dargestellt.

Vorteile von Netzwerken mit einer Domäne:

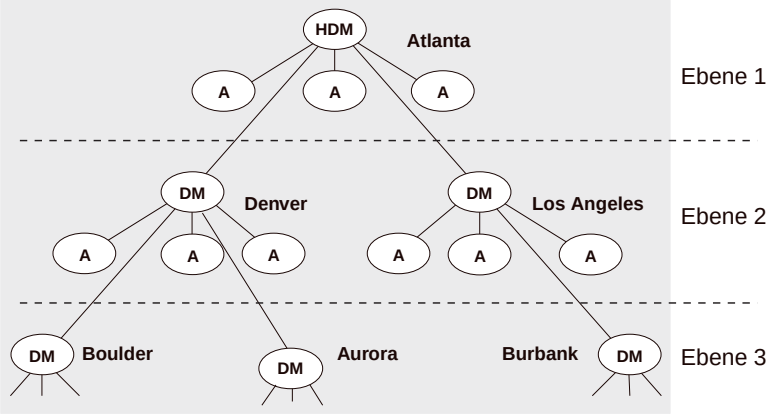
- Eine einfachere Architektur.
- Zentrale Steuerung und zentrales Management.

Nachteile von Netzwerken mit einer Domäne:

- Die gesamte Kommunikation wird über den Hauptdomänenmanager geführt. Dies kann bei weitläufigen Netzwerken zu einer hohen Netzwerkauslastung führen.
- Ein Ausfall des Hauptdomänenmanagers wirkt sich auf das gesamte TWS-Netzwerk aus.

Netzwerke mit mehreren Domänen

Netzwerke mit mehreren Domänen eignen sich vor allem für Unternehmen mit mehreren Standorten, Abteilungen oder Geschäftsfunktionen. Ein TWS-Netzwerk mit mehreren Domänen besteht aus einem Hauptdomänenmanager, einer beliebigen Anzahl an untergeordneten Domänenmanagern und einer beliebigen Anzahl an Agenten in jeder Domäne. Die Agenten kommunizieren nur mit ihren jeweiligen Domänenmanagern, und diese wiederum nur mit ihren übergeordneten Domänenmanagern.



Dieses Diagramm stellt ein Beispiel für ein Netzwerk mit mehreren Domänen dar. Es basiert auf dem Beispiel für Netzwerke mit nur einer Domäne. Der Hauptdomänenmanager befindet sich in Atlanta. Er enthält die Datenbankdateien für die Dokumentierung der Zeitplanungsobjekte und verteilt die Symphony-Datei an seine Agenten sowie an die Domänenmanager in Denver und Los Angeles. Die Domänenmanager in Denver und Los Angeles verteilen die Symphony-Datei anschließend an ihre jeweiligen Agenten sowie an untergeordnete Domänenmanager in Boulder, Aurora und Burbank. Der Hauptdomänenmanager in Atlanta ist für die Übermittlung von domänenübergreifenden Informationen innerhalb des Netzwerks zuständig.

Die gesamte Kommunikation mit dem Domänenmanager in Boulder wird über den übergeordneten Domänenmanager in Denver geleitet. Sind in der Domäne Boulder Pläne oder Jobs vorhanden, die von Plänen oder Jobs in der Domäne Aurora abhängig sind, werden diese Abhängigkeiten vom Domänenmanager in Denver aufgelöst. Die meisten Abhängigkeiten zwischen den Agenten werden lokal von den untergeordneten Domänenmanagern verarbeitet, wodurch der Datenaustausch auf dem Weitverkehrsnetzwerk (Wide Area Network, WAN) erheblich reduziert wird.

Vorteile von Netzwerken mit mehreren Domänen:

- Ad-hoc-Zeitpläne und -Jobs können vom Hauptdomänenmanager oder von jedem anderen Domänenmanager oder Agenten übergeben werden, der Zugriff auf die Datenbankdateien auf dem Hauptdomänenmanager hat.
- Verringerter Datenaustausch auf dem Netzwerk aufgrund der begrenzten Verarbeitung
- Lokale Auflösung von Abhängigkeiten zwischen Agenten
- Eingegrenztes Sicherheitsmanagement und eingegrenzte Verwaltung
- Flexible Topologie, die Ihrem Geschäftsmodell entspricht
- Nahtloser Wechsel zum Sicherungsdomänenmanager

Zu einem Sicherungsdomänenmanager wechseln

Jede Domäne hat einen Domänenmanager und optional einen oder mehrere Sicherungsdomänenmanager. Ein Sicherungsdomänenmanager muss sich in derselben Domäne befinden wie der Domänenmanager, als dessen Sicherung er dient. Bei einem Sicherungsdomänenmanager muss es sich um einen fehlertoleranten Agenten handeln, auf dem dieselbe Version des Produkts ausgeführt wird wie auf dem Domänenmanager, der ersetzt werden soll; in seiner Workstationdefinition müssen die Optionen **Abhängigkeiten auflösen** und **Vollständiger Status** aktiviert sein.

Wenn im Laufe des Arbeitstages ein Domänenmanager ausfällt, können Sie über die Job Scheduling Console oder mit dem Befehl **switchmgr** an der Befehlszeile von Console Manager (conman) zu einem Sicherungsdomänenmanager wechseln. Ein Managerwechsel kann von jedem Benutzer durchgeführt werden, der berechtigt ist, die Workstations mit Domänenmanagern und Sicherungsdomänenmanagern zu starten bzw. zu stoppen.

Beim Wechseln eines Managers wird zunächst der Sicherungsdomänenmanager gestoppt und anschließend als neuer Domänenmanager erneut gestartet; der alte Domänenmanager wird in einen fehlertoleranten Agenten umgewandelt. Die Identifikationen der aktuellen Domänenmanager werden mit Hilfe der Symphony-Datei von einem Verarbeitungstag zum nächsten übernommen, so dass jeder Wechsel so lange gültig bleibt, bis erneut zum eigentlichen Domänenmanager gewechselt wird.

Erweiterte Datenbank und lange Objektnamen

Die TWS-Datenbanken sowie die Symphony-Datei wurden erweitert, damit mehr Datensätze und längere Objektnamen gespeichert werden können. Dies hat eine erhöhte Flexibilität bei der Benennung von Zeitplanungsobjekten zur Folge, besonders in Zusammenhang mit Paketen wie z. B. SAP R/3, Baan usw.

- In Tivoli Workload Scheduler werden erweiterte Datenbanken standardmäßig verwendet. Sie ermöglichen die Verwendung von langen Namen für Zeitplanungsobjekte; daher können bis zu 40 Zeichen bei der Benennung von Jobs verwendet werden, im Gegensatz zu den acht Zeichen in früheren Versionen.
- Nicht erweiterte Datenbanken werden noch unterstützt und müssen auf MPE-Agenten (HP3000) verwendet werden.

  Anmerkung: Stellen Sie sicher, dass auf dem Master keine neuere Scheduler-Version als auf den fehlertoleranten Agenten in der Domäne ausgeführt wird.

Datenbankerstellung

Bei Erstinstallationen werden die Tivoli Workload Scheduler-Datenbanken auf dem Hauptdomänenmanager erstellt, wenn Sie die Tivoli Workload Scheduler-Steuerkomponente zum ersten Mal ausführen. Sie können feststellen, ob es sich dabei um erweiterte oder nicht erweiterte Datenbanken handelt, indem Sie die globale Option **expanded version = [yes|no]** überprüfen. Diese Option wird bei der Installation der Tivoli Workload Scheduler-Steuerkomponente vom Anpassungsskript gesetzt. Die Standardeinstellung ist **yes** (erweitert).

Bei der ersten Installation von Tivoli Workload Scheduler können Sie sich für nicht erweiterte Datenbanken entscheiden und diese zu einem späteren Zeitpunkt erweitern. Führen Sie den Befehl **dbexpand** auf dem Hauptdomänenmanager aus, um die Datenbanken zu erweitern. Das Programm setzt die globale Option **expanded version** auf **yes**, erstellt Sicherungskopien der vorhandenen Datenbanken und erweitert die Datenbanken, so dass sie lange Objektnamen unterstützen. Führen Sie **dbexpand** nur einmal aus, da sonst die Sicherungskopie der vorhandenen Datenbanken gelöscht wird. Die Sicherungskopien werden in einem Verzeichnis namens **mozart.old** innerhalb des Verzeichnisses **mozart** abgelegt.

Workstationnamen

Job Scheduling in einem TWS-Netzwerk ist auf mehrere Computer verteilt. Jeder Computer verfügt über einen eindeutigen Workstationnamen, damit eine korrekte Überwachung von Jobs, Zeitplänen und anderen Objekten möglich ist. Bei den Namen kann es sich auch um Netzwerkknotennamen handeln; es muss nur gewährleistet sein, dass folgende Namenskonventionen von TWS eingehalten werden:

- In nicht erweiterten Netzwerken beträgt die maximale Länge von Workstationnamen acht Zeichen. Die Namen müssen mit einem Buchstaben beginnen und können aus alphanumerischen Zeichen, Bindestrichen und Unterstreichungszeichen bestehen.
- Bei erweiterten Netzwerken können die Namen aus bis zu 16 alphanumerischen Zeichen, Bindestrichen und Unterstreichungszeichen bestehen. Beim ersten Zeichen muss es sich um einen Buchstaben handeln.

Übersicht über die Installation

Nachfolgend finden Sie eine Zusammenfassung des Installationsprozesses für Tivoli Workload Scheduler. Dieses Handbuch enthält detaillierte Anweisungen zu jedem Schritt.

1. Gehen Sie auf allen Domänenmanagern (einschließlich Haupt- und Sicherungsdomänenmanagern) in der folgenden Reihenfolge vor:

   a. Wenn Sie eine vorhandene Version von Maestro oder Tivoli Workload Scheduler aktualisieren, trennen Sie die Verbindungen zu den anderen Workstations in der Domäne, und stoppen Sie Maestro bzw. Tivoli Workload Scheduler. Siehe „Tivoli Workload Scheduler trennen und stoppen” auf Seite 20 für NT und „Tivoli Workload Scheduler aktualisieren” auf Seite 40 für UNIX.

      Installieren Sie Tivoli Workload Scheduler. Siehe Kapitel 2, „Unter Windows NT installieren” auf Seite 19 oder Kapitel 3, „Unter UNIX installieren” auf Seite 35. Sie können diesen Schritt auch nach dem nächsten ausführen.

   b. Installieren Sie Tivoli Management Framework. Wenn Sie innerhalb Ihres Unternehmens nicht bereits über einen Tivoli-Verwaltungsbereich (Tivoli Management Region - TMR) verfügen, installieren Sie TMF als TMR-Server. Ist bereits ein TMR vorhanden, können Sie TMF auch als verwalteten Knoten installieren. Nähere Informationen hierzu finden Sie in der Installationsdokumentation von TMF.

c. Installieren Sie die Job Scheduling Services und den Connector in Tivoli Management Framework. Siehe das *Tivoli Workload Scheduler Console Benutzerhandbuch*.

d. Erstellen Sie auf der Tivoli-Arbeitsoberfläche einen Tivoli-Administrator für Tivoli Workload Scheduler. Fügen Sie also entweder den Benutzer **TWS** oder den Benutzer **Maestro** als Anmeldenamen für den TMF-Administrator hinzu, oder erstellen Sie einen neuen TMF-Administrator auf Grundlage der Anmeldenamen **TWS** oder **Maestro**.

e. Fügen Sie den Tivoli-Administrator den Tivoli Workload Scheduler-Sicherheitsinformationen hinzu.

- Unter NT müssen Sie Tivoli Workload Scheduler stoppen, bevor Sie den Befehl *makesec* ausführen können.
- Unter UNIX können Sie den Befehl *makesec* ausführen, ohne vorher Tivoli Workload Scheduler zu stoppen; Sie müssen Tivoli Workload Scheduler jedoch stoppen und erneut starten, damit die Sicherheitsfunktion aktiviert wird.

f. Installieren Sie die Job Scheduling Console. Siehe das *Tivoli Workload Scheduler Console Benutzerhandbuch*.

2. Installieren Sie folgende Komponenten auf jeder Workstation, die als Agent im Tivoli Workload Scheduler-Netzwerk dienen soll:
   - Tivoli Workload Scheduler
   - Job Scheduling Console (optional). Sie können die Job Scheduling Console auch auf Workstations installieren, die sich zwar nicht im Tivoli Workload Scheduler-Netzwerk befinden, jedoch über TCP/IP mit einem Domänenmanager verbunden sind, auf dem der Connector ausgeführt wird.
3. Melden Sie sich an der Job Scheduling Console an, und definieren Sie die Workstations des Tivoli Workload Scheduler-Netzwerks in der Scheduler-Datenbank.
4. Jetzt können Sie mit der Job Scheduling Console Jobs, Jobnetze und andere Zeitplanungsobjekte definieren.

Produktgruppen

Tivoli Workload Scheduler ist in Produktgruppen gegliedert. Dies ermöglicht die Installation mehrerer Kopien eines Produkts auf einem einzelnen Computer, indem für jede Kopie eine andere Gruppe zugeordnet wird.

Komponentendatei

Die Produktgruppen werden in der Komponentendatei definiert. Ist diese Datei vor der Installation nicht vorhanden, wird sie vom Skript *customize* unter UNIX bzw. vom Programm *Setup* unter Windows NT erstellt. Hier ein Beispiel für UNIX:

```
<product>       <version>      <home directory>                    <product group>
Maestro         6.1            /opt/maestro                        production
Netman          3.3.5          /opt/netman                         production
```

Die Einträge in der Datei werden von **customize** bzw. **Setup** automatisch erstellt und aktualisiert. Die Produktgruppe *workstation_user* wird unter Windows NT standardmäßig verwendet, wenn Sie nicht eine andere Gruppe angeben.

Unter UNIX wird der Dateiname der Komponentendatei in folgender Variable definiert:

```
UNISON_COMPONENT_FILE
```

Ist für diese Variable kein Wert gesetzt, verwendet das Skript **customize** den folgenden Dateinamen:

```
/usr/unison/components
```

Unter Windows NT wird der Name der Komponentendatei, netmanhome\components, bei der ersten Installation dieser Version von Workload Scheduler in der Registrierungsdatenbank eingetragen. Siehe „Basisverzeichnis von Netman”.

Die Komponentendatei anzeigen

Nach der Installation bzw. der Aktualisierung können Sie den Inhalt der Komponentendatei anzeigen, indem Sie das Programm **ucomp** wie folgt ausführen:

```
ucomp -l
```

Basisverzeichnis von Netman

Mit der Option *-nmhome* im Skript **customize** unter UNIX können Sie ein Basisverzeichnis für Netman angeben. Wird dieser Schritt nicht ausgeführt, ist die Standardeinstellung unter UNIX das Basisverzeichnis von Workload Scheduler.

Lokale Netman-Optionen

Wenn Netman ein anderes Basisverzeichnis als Tivoli Workload Scheduler hat, werden die folgenden lokalen Optionen von der Workload Scheduler-Datei **localopts** in eine separate Datei **localopts** im Verzeichnis von Netman verschoben:

```
nm mortal
nm port
nm read
nm retry
merge stdlists
stdlist width
syslog local
```

Folgende Option kann bei Bedarf hinzugefügt werden:

```
nm ipvalidate
```

Weitere Informationen zu lokalen Optionen finden Sie im *Tivoli Workload Scheduler Console Benutzerhandbuch*.

Nach der Installation

Lesen Sie Folgendes nach der Installation von Tivoli Workload Scheduler:

- Kapitel 4, „Sicherheit definieren” auf Seite 49 zum Setzen der Tivoli Workload Scheduler-Sicherheitsparameter
- Kapitel 5, „Optionale Anpassungsoptionen” auf Seite 69 zum Setzen der lokalen und globalen Optionen
- Anhang A, „Zeitzonen und Prüfung” auf Seite 89 mit Informationen zu diesen Optionen

2

Unter Windows NT installieren

In diesem Kapitel wird der Installationsprozess von Tivoli Workload Scheduler für Windows NT-Systeme und -Netzwerke beschrieben.

Systemvoraussetzungen

Nachfolgend finden Sie die Systemvoraussetzungen für Tivoli Workload Scheduler auf Windows NT-Systemen:

- Windows NT Version 4.0 mit Service Pack 5 oder 6a
- CD-ROM-Laufwerk für die Installation
- Eine NTFS-Partition für die Installation von Tivoli Workload Scheduler
- Ca. 100 MB freier Plattenspeicherplatz für Domänenmanager und fehlertolerante Agenten, die als Sicherungsmaster konfiguriert sind. Ca. 40 MB für Standardagenten. Tivoli Workload Scheduler erstellt außerdem Protokolldateien und temporäre Dateien, die auf dem lokalen Festplattenlaufwerk gespeichert werden. Der benötigte Plattenspeicherplatz hängt von der Anzahl der von Tivoli Workload Scheduler verwalteten Jobs ab sowie davon, wie lange die Protokolldateien gespeichert bleiben sollen.
- 128 MB Arbeitsspeicher und 128 MB Auslagerungsspeicher für Domänenmanager und fehlertolerante Agenten. Der Bedarf von Standardagenten ist geringer.
- TCP/IP-Netzwerkkommunikation
- Für die ordnungsgemäße Installation ist ein Tivoli Workload Scheduler-Benutzerkonto erforderlich. Dieses Benutzerkonto können Sie entweder vor der Installation einrichten oder vom Setup-Programm erstellen lassen.

  Anmerkung: Wenn Sie einen Benutzer erstellen, bevor Sie das Setup-Programm ausführen, wählen Sie einen Benutzernamen aus, in dem kein Punkt (.) enthalten ist. Benutzerkonten, die Punkte enthalten, sind für MPE-Benutzer reserviert. Näheres hierzu finden Sie unter „Das Tivoli Workload Scheduler-Konto manuell erstellen” auf Seite 31.

Bevor Sie fortfahren, lesen Sie die *Tivoli Workload Scheduler Release-Informationen* mit aktuellen Informationen zu den Systemvoraussetzungen und zu plattformspezifischen Themen.

Für Endpunkt-zu-Endpunkt-Zeitplanung installieren

Wenn Sie Tivoli Workload Scheduler auf einer Workstation installieren, die als ein verteilter Agent für die Endpunkt-zu-Endpunkt-Zeitplanung (ein Domänenmanager, ein fehlertoleranter Agent oder ein Standardagent) verwendet wird, müssen Sie Folgendes ausführen:

1. Geben Sie OPCMASTER als Name des Hauptdomänenmanagers im Feld **Master-CPU** des Fensters **Konfigurationsdaten** an. Eine Beschreibung des Fensters finden Sie unter Schritt 12.
2. Definieren Sie nach der Installation die WEZ-Zeitzone für die Workstation. Dies ist notwendig, damit Start- und Endzeiten auf dem Tivoli Workload Scheduler for z/OS-Controller mit der lokalen Controller-Zeitzone gemeldet und die Zeitabhängigkeiten richtig aufgelöst werden können. Weitere Informationen hierzu finden Sie unter „Zeitzonen” auf Seite 89.

Aktualisierungen vorbereiten

Führen Sie vor der Ausführung des Setup-Programms die nachfolgenden Schritte aus, wenn Sie eine vorhandene Installation von Tivoli Workload Scheduler aktualisieren wollen. Lesen Sie nach der Installation den Abschnitt „Eine Aktualisierung abschließen” auf Seite 28, in dem wichtige Schritte, die nach der Installation ausgeführt werden müssen, beschrieben werden.

Anmerkung: Normalerweise werden bei der Aktualisierung eines Tivoli Workload Scheduler-Netzwerks zunächst die fehlertoleranten Agenten und die Standardagenten aktualisiert und anschließend der Hauptdomänenmanager. Lesen Sie aber vor der Aktualisierung Ihres Tivoli Workload Scheduler-Netzwerks die *Tivoli Workload Scheduler Release-Informationen* mit den neuesten Informationen zur Migration auf eine neue Softwareversion.

Tivoli Workload Scheduler trennen und stoppen

Sie müssen die Verbindung zu Workstations unterbrechen und die Tivoli Workload Scheduler-Prozesse stoppen, bevor Sie die Aktualisierung Ihrer Installation starten. Führen Sie folgende Schritte aus:

1. Unterbrechen Sie über die Job Scheduling Console die Verbindung der Zielworkstation mit den anderen Workstations im Netzwerk. Sie können auch an der Befehlszeile den folgenden Befehl eingeben:

   ```
   conman "unlink workstationname;noask"
   ```

2. Stoppen Sie die Zielworkstation über die Job Scheduling Console, und fahren Sie sie herunter. Geben Sie an der Befehlszeile den folgenden Befehl ein:

   ```
   conman "stop"
   conman "shut;wait"
   ```

Sie können dieselbe Task auch durch die Ausführung einer der folgenden zwei Aktionen erfüllen:

1. Beenden Sie den Prozess über den Windows NT Task-Manager.
2. Führen Sie den folgenden Befehl in einem MS-DOS-Eingabeaufforderungsfenster aus:

   ```
   maestrohome/shutdown
   ```

Sicherungsdateien

Das Setup-Programm von Tivoli Workload Scheduler versetzt Ihre Kopien der Dateien **Sfinal**, **Jnextday**, **NetConf**, **StartUp** sowie das Verzeichnis **methods** in ein Verzeichnis mit dem folgenden Namen

TWShome\config.old

Diese Dateien werden häufig angepasst, um den besonderen Anforderungen des Benutzers zu entsprechen; Sie können mit den gesicherten Kopien nach der Aktualisierung Ihre Änderungen an den Dateien übernehmen. Das Setup-Programm überschreibt keine Dateien in den Verzeichnissen **mozart**, **stdlist** oder **unison**, die nach der Installation von Tivoli Workload Scheduler geändert wurden.

Wenn Sie andere Dateien während des Aktualisierungsprozesses schützen möchten, sollten Sie diese jetzt kopieren oder umbenennen. Vorsorglich sollten Sie eine Sicherungskopie des Verzeichnisses *TWShome* erstellen.

Sie können jetzt das Setup-Programm ausführen. Siehe „Die Software installieren".

Die Software installieren

Die Tivoli Workload Scheduler-Steuerkomponente wird unter Windows NT mit Hilfe des Setup-Programms installiert.

Das Setup-Programm

Das Setup-Programm installiert bzw. aktualisiert Tivoli Workload Scheduler. Folgende Funktionen werden durchgeführt:

- Tivoli Workload Scheduler-Erstinstallation: Tivoli Workload Scheduler und Netman installieren. Eine Komponentendatei mit neuen Einträgen erstellen.
- Erstmalige Aktualisierung von Maestro: Tivoli Workload Scheduler und, falls erforderlich, Netman aktualisieren. Eine Komponentendatei mit neuen Einträgen erstellen.
- Nachfolgende Aktualisierungen von Tivoli Workload Scheduler: Tivoli Workload Scheduler und, falls erforderlich, Netman aktualisieren. Einträge in der Komponentendatei aktualisieren.

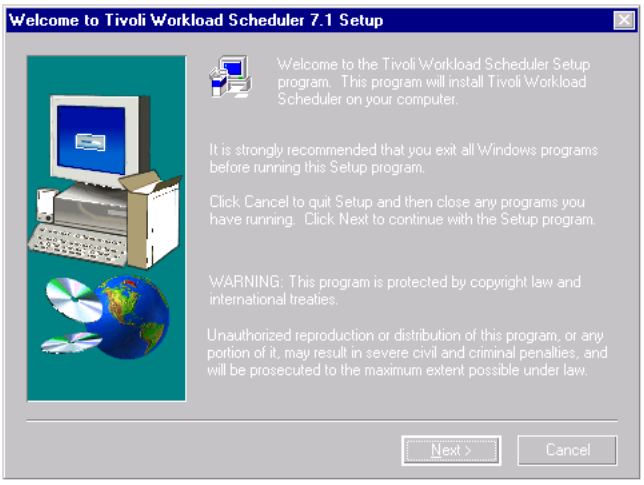
Das Setup-Programm ausführen

1. Stellen Sie sicher, dass Sie auf der Workstation, auf der Sie Tivoli Workload Scheduler installieren wollen, über lokale Administratorberechtigungen verfügen.
2. Schließen Sie alle offenen Windows-Anwendungen einschließlich des Windows NT-Explorers.
3. Legen Sie die Tivoli Workload Scheduler-CD in das Laufwerk ein.
4. Führen Sie das Setup-Programm von Tivoli Workload Scheduler aus:

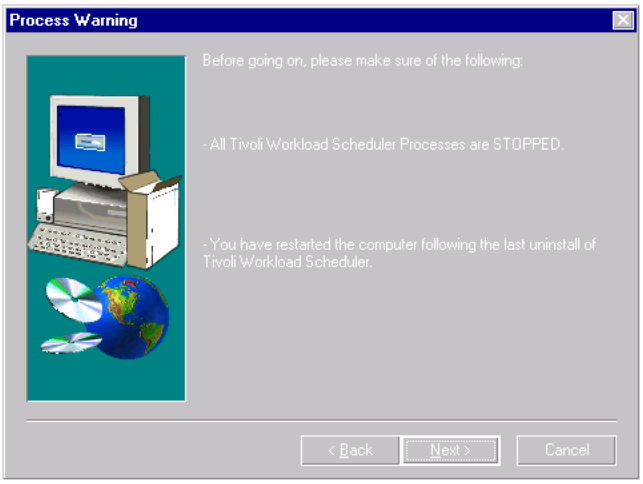
   d:\TWS\i386\maestro\setup.exe

   Dabei steht *d* für den Laufwerkbuchstaben Ihres CD-ROM-Laufwerks.
5. Das Fenster **Wählen Sie eine Setup-Sprache aus** wird angezeigt. Wählen Sie die gewünschte Installationssprache aus, und klicken Sie **OK** an.

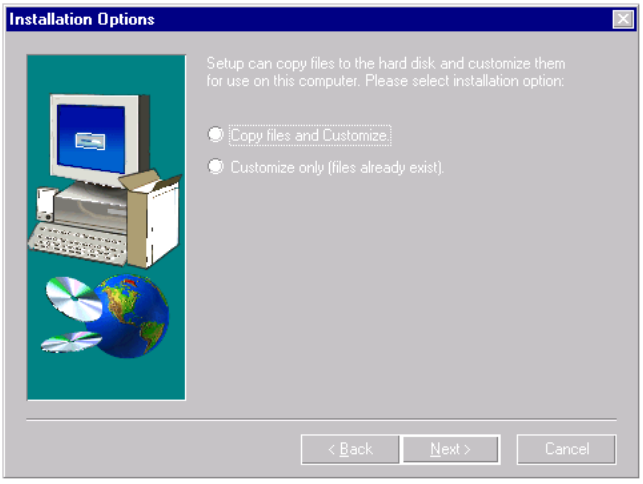
6. Das Begrüßungsfenster wird angezeigt.


Welcome to Tivoli Workload Scheduler 7.1 Setup
Welcome to the Tivoli Workload Scheduler Setup program. This program will install Tivoli Workload Scheduler on your computer.
It is strongly recommended that you exit all Windows programs before running this Setup program.
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.
WARNING: This program is protected by copyright law and international treaties.
Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.
Next >
Cancel

7. Klicken Sie im Begrüßungsfenster **Weiter>** an, um fortzufahren. Das Fenster **Prozesswarnung** wird angezeigt. Klicken Sie **Weiter>** an, um fortzufahren. Wenn Sie eine Aktualisierung von Tivoli Workload Scheduler durchführen, stellen Sie sicher, dass alle Tivoli Workload Scheduler-Prozesse einschließlich der Tivoli-Dienste gestoppt sind. Wenn Sie Tivoli Workload Scheduler vor der Ausführung des Setup-Programms deinstalliert haben, müssen Sie den Computer erneut starten, bevor Sie fortfahren (Hinweise zur Deinstallation finden Sie unter „Tivoli Workload Scheduler deinstallieren” auf Seite 33). Beenden Sie Setup dazu, indem Sie **Abbrechen** anklicken.


Process Warning
Before going on, please make sure of the following:
- All Tivoli Workload Scheduler Processes are STOPPED.
- You have restarted the computer following the last uninstall of Tivoli Workload Scheduler.
< Back
Next >
Cancel

8. Wählen Sie im Fenster **Installationsoptionen** die Installationsart aus, und klicken Sie **Weiter>** an.


Installation Options
Setup can copy files to the hard disk and customize them for use on this computer. Please select installation option:
Copy files and Customize.
Customize only (files already exist).
< Back
Next >
Cancel

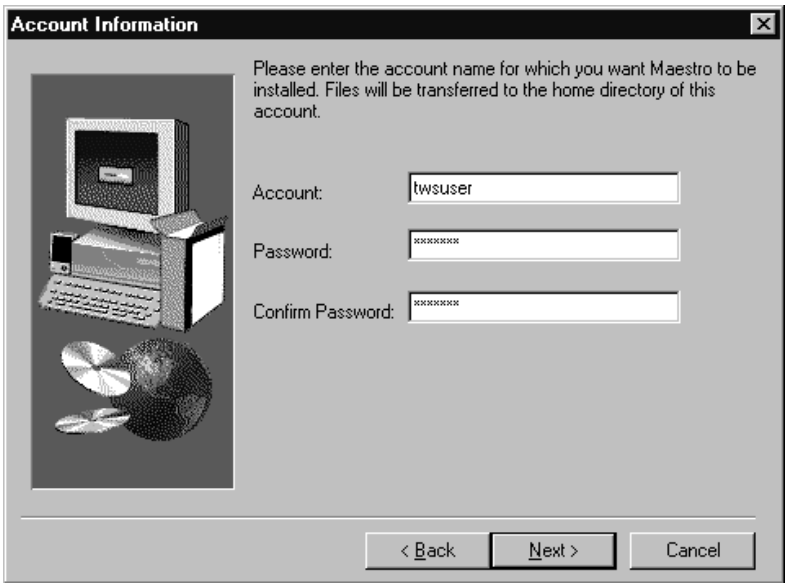
Dateien kopieren und anpassen

Die Netman-Dateien werden von der CD kopiert, und die Installation wird angepasst. Diese Option muss sowohl bei einer Neuinstallation als auch bei einer Aktualisierung ausgewählt werden.

Nur anpassen

Die vorhandenen Dateien werden angepasst. Dadurch werden die Dateiberechtigungen auf die Standardeinstellungen zurückgesetzt.

9. Geben Sie die folgenden Informationen im Fenster **Benutzerinformationen** an, und klicken Sie **Weiter>** an:


Account Information
Please enter the account name for which you want Maestro to be installed. Files will be transferred to the home directory of this account.
Account:
twsuser
Password:
Confirm Password:
< Back
Next >
Cancel

Konto

Der Name des Tivoli Workload Scheduler-Benutzers (in der Regel *twsuser* oder *maestro*). Wenn Sie einen anderen Namen als *twsuser* oder *maestro* angeben, stellen Sie sicher, dass er keinen Punkt (.) enthält, dieses Zeichen ist für MPE-Benutzernamen reserviert.

Die Software wird im angegebenen Verzeichnis installiert bzw. aktualisiert. Sie können gegebenenfalls die Domäne angeben, z. B. *hqtrs\twsuser*. Wenn Sie die Domäne nicht angeben, sucht das Setup-Programm nach einem passenden Konto: 1. lokal, 2. in der Domäne oder 3. in einer gesicherten Domäne.

Setup erstellt automatisch das Konto und gegebenenfalls das Basisverzeichnis und erteilt die erforderlichen Berechtigungen.

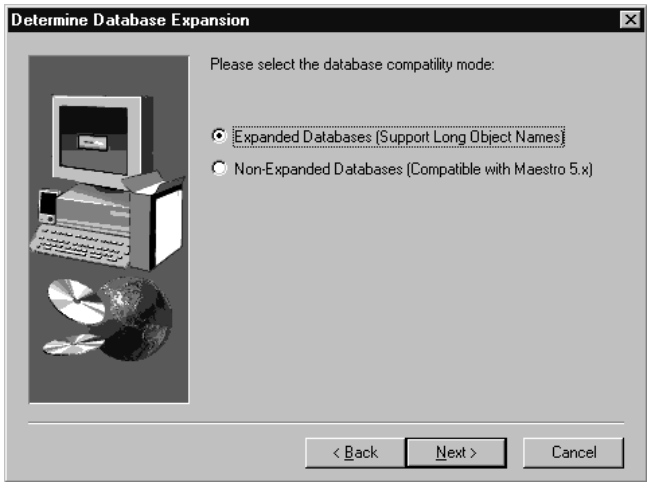
Kennwort
Das Kennwort des Kontos.

Kennwort bestätigen
Geben Sie das Kennwort des Kontos erneut ein.

Anmerkung: Wenn Sie mehr als eine Instanz von Tivoli Workload Scheduler auf demselben Computer installieren, sollten Sie berücksichtigen, dass bei jeder Installation versucht wird, Dateien in das Verzeichnis *TWShome*\..\unison zu kopieren. Geben Sie unterschiedliche übergeordnete Verzeichnisse an, um Konflikte zu vermeiden. Beispiel: *c:\apps\maestro* und *c:\apps1\maestro*.

10. Geben Sie das Zielverzeichnis an, in dem Tivoli Workload Scheduler installiert werden soll. Möchten Sie ein anderes Verzeichnis als das Standardverzeichnis verwenden, klicken Sie **Durchsuchen** an, um ein anderes Verzeichnis anzugeben. Wenn Sie fertig sind, klicken Sie **Weiter** an, um fortzufahren.

11. Wählen Sie im Fenster **Datenbankerweiterung feststellen** eine der möglichen Optionen aus, und klicken Sie anschließend **Weiter>** an. Dieses Fenster wird nicht angezeigt, wenn Sie eine vorhandene Installation aktualisieren.

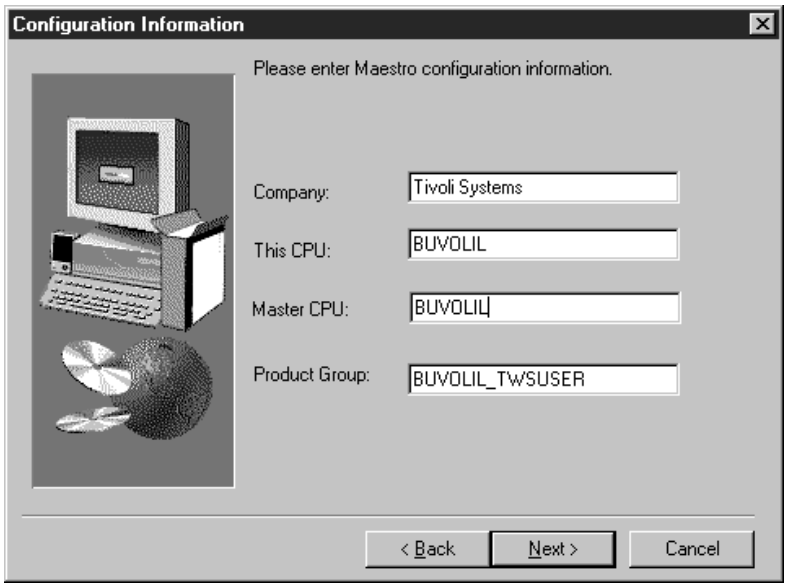

Determine Database Expansion
Please select the database compatility mode:
Expanded Databases (Support Long Object Names)
Non-Expanded Databases (Compatible with Maestro 5.x)
< Back
Next >
Cancel

Erweiterte Datenbanken (Unterstützung langer Objektnamen)
Es werden erweiterte Datenbanken verwendet, die nur in Maestro 6.0 und in Tivoli Workload Scheduler verfügbar sind.

Nicht erweiterte Datenbanken (Kompatibel mit Maestro 5.x)
Es werden nicht erweiterte Datenbanken verwendet. Diese Auswahl sollte getroffen werden, wenn Sie Tivoli Workload Scheduler auf dem Hauptdomänenmanager eines Netzwerks mit Workstations installieren, auf denen eine beliebige Version von Maestro for MPE ausgeführt wird. Die Datenbanken können mit Hilfe des Befehls **dbexpand** zu einem späteren Zeitpunkt erweitert werden. Im Handbuch *Tivoli Workload Scheduler Referenz* finden Sie weitere Informationen zum Befehl **dbexpand**.

Anmerkung: Wenn sich in Ihrem Netzwerk MPE-Workstations befinden, müssen Sie den **nicht erweiterten** Modus verwenden. MPE-Workstations können nicht in **erweiterte** Tivoli Workload Scheduler-Netzwerke aufgenommen werden.

12. Geben Sie die folgenden Informationen im Fenster **Konfigurationsdaten** an, und klicken Sie **Weiter>** an:



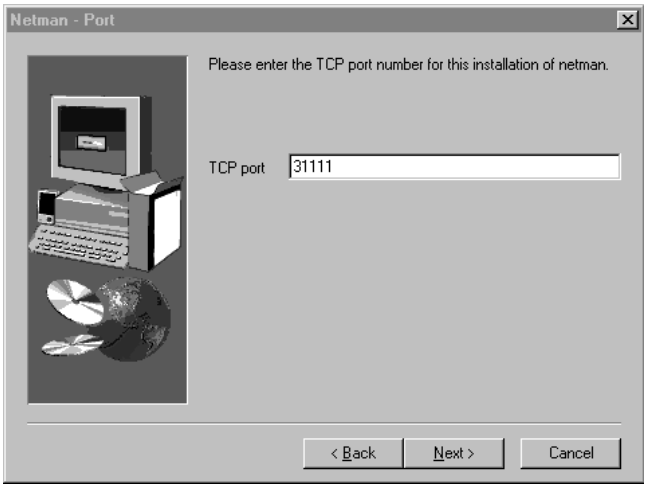
Firma Der Name Ihres Unternehmens.

Diese CPU
Der Tivoli Workload Scheduler-Name dieser Workstation. Bei der nicht erweiterten Version von Tivoli Workload Scheduler kann es sich hierbei um bis zu acht alphanumerische Zeichen, Bindestriche (-) oder Unterstreichungszeichen (_) handeln. Das Anfangszeichen muss ein Buchstabe sein. Bei der erweiterten Version können bis zu 16 Zeichen eingegeben werden. Mit diesem Namen wird zu einem späteren Zeitpunkt die Workstation in Tivoli Workload Scheduler formal definiert.

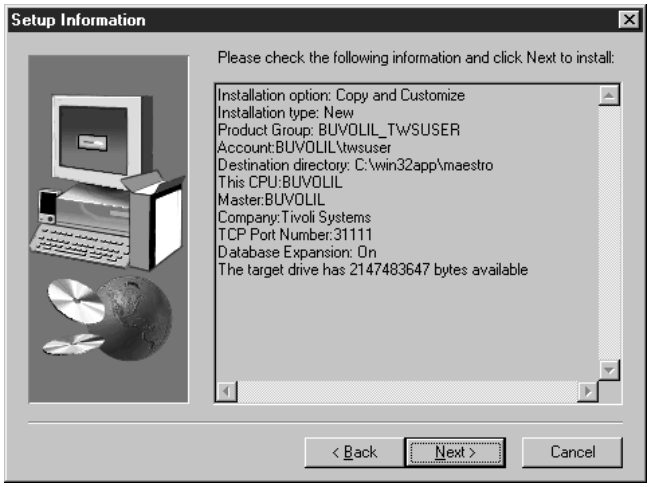
Master-CPU
Der Tivoli Workload Scheduler-Name des Hauptdomänenmanagers. Wenn es sich hier um den Hauptdomänenmanager oder um eine eigenständige Workstation handelt, entspricht dieser Name dem Namen im Feld **Diese CPU**. Bei der nicht erweiterten Version von Tivoli Workload Scheduler kann es sich hierbei um bis zu acht alphanumerische Zeichen, Bindestriche (-) oder Unterstreichungszeichen (_) handeln. Das Anfangszeichen muss ein Buchstabe sein. Bei der erweiterten Version können bis zu 16 Zeichen eingegeben werden. Wenn Sie die Installation auf dem Hauptdomänenmanager durchführen, erstellt das Setup-Programm automatisch dessen Workstationdefinition.

Produktgruppe
Der Name der Produktgruppe für diese Installation oder diese Aktualisierung. Die Standardeinstellung ist *"workstation_user"*.

13. Geben Sie im Fenster **Netman - Anschluss** die Netman-Anschlussnummer an, und klicken Sie **Weiter>** an. Dabei muss es sich um einen 16-Bit-Wert ohne Vorzeichen im Bereich von 1 bis 65535 handeln (Beachten Sie, dass die Werte zwischen 0 und 1023 für Standarddienste, wie z. B. FTP, TELNET, HTTP usw., reserviert sind.). Der Standardwert ist 31111.



14. Zum Schluss werden im Fenster **Installationsdaten** die Informationen zusammengefasst, die Setup bei der Installation von Tivoli Workload Scheduler verwendet. Klicken Sie **<Zurück** an, um die Informationen zu prüfen oder zu ändern, oder klicken Sie **Weiter>** an, um Tivoli Workload Scheduler zu installieren bzw. zu aktualisieren und das Programm zu verlassen.



15. Die Installation der Tivoli Workload Scheduler-Steuerkomponente ist nun abgeschlossen.

Nach der Installation bzw. der Aktualisierung

Führen Sie nach der Installation oder der Aktualisierung von Tivoli Workload Scheduler folgende Tasks aus:

Eine Aktualisierung abschließen

Gehen Sie wie folgt vor, wenn Sie eine Aktualisierung einer bestehenden Installation vornehmen:

1. Verwenden Sie die Dateien, die Sie zuvor gesichert haben (siehe „Aktualisierungen vorbereiten” auf Seite 20) als Grundlage, um die neuen Versionen der Dateien auf jeder aktualisierten Workstation entsprechend zu ändern.
2. Starten Sie Tivoli Workload Scheduler auf dem Hauptdomänenmanager erneut.

Die Aktualisierung von Tivoli Workload Scheduler ist nun abgeschlossen.

Die Variable PATH setzen

Melden Sie sich als Benutzer der Gruppe der lokalen Administratoren bzw. der Domänenadministratoren an. Fügen Sie in der Systemumgebung die Verzeichnisse *TWShome* und *TWShome*\bin der PATH-Variablen hinzu. Beispiel:

```
andere_verzeichnisse;C:\win32app\maestro;C:\win32app\maestro\bin
```

Verzeichnisse und Dienste von Tivoli Workload Scheduler

Setup installiert die Tivoli Workload Scheduler-Dateien in *TWShome* und in *TWShome***\..\unison**, das auch von anderen Tivoli-Produkten verwendet wird. Ist das Basisverzeichnis *(TWShome)* beispielsweise **c:\win32app\maestro**, wird auch das Verzeichnis **c:\win32app\unison** erstellt.

Setup registriert auch die folgenden Dienste beim Dienststeuerungsmanager:

- Tivoli MAESTRO für die Benutzer (*user*)
- Tivoli Netman für die Workstationbenutzer (*wkstn_user*)
- Tivoli Token-Dienst für die Benutzer (*user*)

Beachten Sie, dass der Dienststeuerungsmanager über eine eigene Datenbank zur Verwaltung von Benutzerkennwörtern verfügt. Wird nach der Installation also das Kennwort von *TWSuser* geändert, müssen Sie mit dem Applet **Dienste** in der Systemsteuerung das neue Kennwort für den Token-Dienst und Netman zuweisen.

Der Tivoli Workload Scheduler-Dienst

Der Jobman-Dienst wird unter dem Konto SYSTEM ausgeführt und verfügt über die Berechtigung ″Interaktive Beziehung mit Desktop erlauben″. Diese Berechtigung können Sie aus Sicherheitsgründen entziehen. Dadurch kann der Dienst jedoch keine interaktiven Jobs mehr starten, die in einem Fenster auf der Arbeitsoberfläche des Benutzers ausgeführt werden. Auf diese Jobs kann nicht zugegriffen werden, und sie haben keinen Zugriff auf Arbeitsoberflächenressourcen. Daher werden sie möglicherweise entweder endlos ausgeführt oder auf Grund eines Ressourcenmangels abnormal beendet.

Die dezentrale Verwaltung einrichten

Die Verwaltung der Tivoli Workload Scheduler-Zeitplanungsobjekte kann nicht nur vom Tivoli Workload Scheduler-Hauptdomänenmanager aus erfolgen (auf diesem sind die Datenbanken vorhanden), sondern auch von jedem anderen Computer aus.

Damit Sie die Zeitplanungsobjekte auf diese Weise verwalten können, müssen Sie die Verzeichnisse des Hauptdomänenmanagers für die gemeinsame Benutzung freigeben und eine Reihe lokaler Optionen auf den anderen Computern definieren.

Die Verzeichnisse des Masters gemeinsam benutzen

Geben Sie die nachfolgenden Verzeichnisse auf dem Hauptdomänenmanager zur gemeinsamen Benutzung frei, und erteilen Sie die notwendigen Berechtigungen, damit die Domänenbenutzer **TWS** oder **maestro** über die vollständige Kontrolle verfügen:

TWShome\mozart
TWShome\..\unison\network

Tivoli Workload Scheduler-Parameter gemeinsam benutzen

Die Substitutionsparameter von Tivoli Workload Scheduler sind in der Regel computerbezogen und werden von daher auf jedem Computer separat verwaltet. Sollen diese Parameter für alle Computer gelten und von einem beliebigen Computer aus verwaltet werden, können Sie entweder das Verzeichnis *TWShome* zur gemeinsamen Benutzung freigeben, wie oben für andere Verzeichnisse beschrieben, oder die Parameterdatenbank bei jeder Änderung vom Hauptdomänenmanager auf die anderen Computer kopieren. Es gibt folgende Datenbankdateien:

TWShome\parameters
TWShome\parameters.KEY

Ein Verzeichnis gemeinsam benutzen

Alternativ zur gemeinsamen Benutzung verschiedener Verzeichnisse können Sie auch alle Datenbankdateien in ein gemeinsames Verzeichnis versetzen, dieses zur gemeinsamen Benutzung freigeben und anschließend in den weiter unten beschriebenen lokalen Optionen den Namen des zur gemeinsamen Benutzung freigegebenen Verzeichnisses angeben. Die Datenbankdateien werden nachfolgend aufgeführt:

In *TWShome*\..\unison\:

```
cpudata
cpudata.KEY
userdata
userdata.KEY
```

In *TWShome*\mozart\:

```
calendars
calendars.KEY
job.sched
job.sched.KEY
jobs
jobs.KEY
mastsked
mastsked.KEY
prompts
prompts.KEY
resources
resources.KEY
```

In *TWShome*:

```
parameters
parameters.KEY
```

Die Dateien werden erstellt, wenn sie von Tivoli Workload Scheduler benötigt werden. Sind sie nicht vorhanden, können Sie in den lokalen Optionen das zur gemeinsamen Benutzung freigegebene Verzeichnis angeben, wie unten beschrieben.

Die lokalen Optionen einstellen

Stellen Sie die lokalen Optionen auf jedem Computer ein, von dem aus Sie die Zeitplanungsobjekte von Tivoli Workload Scheduler verwalten möchten; somit erhalten diese Computer Zugriff auf die zur gemeinsamen Benutzung freigegebenen Master-Datenbanken. Nachfolgend werden die Optionen sowie die Vorgehensweise zu deren Änderung beschrieben.

Beachten Sie, dass jede Option auf einen konventionellen Namen (laufwerk:\ge- meinsames-verzeichnis) oder auf einen Namen nach der allgemeinen Namenskonvention (\\knoten\gemeinsames-verzeichnis) gesetzt werden kann. Wird ein konventioneller Name verwendet, muss der Maestro-Benutzer explizit eine Verbindung zu dem freigegebenen Verzeichnis herstellen. Wird ein Name nach der allgemeinen Namenskonvention verwendet, ist diese explizite Verbindung nicht erforderlich. Es gibt folgende lokale Optionen:

Mozart-Verzeichnis (mozart directory)
: Definiert den Namen des gemeinsam benutzten Verzeichnisses ’mozart’ auf dem Master.

Unison-Netzwerkverzeichnis (unison network directory)
: Definiert den Namen des gemeinsam benutzten Verzeichnisses ’unison’ auf dem Master.

Parameterverzeichnis (parameters directory)
: Definiert den Namen des gemeinsam benutzten Verzeichnisses *TWShome* auf dem Master.

Setzen Sie auf jedem Computer die Optionen wie folgt:

1. Verwenden Sie einen Editor Ihrer Wahl, um die Datei *TWShome*\localopts zu öffnen und zu ändern.
2. Die Optionen sind in der von Tivoli gelieferten Datei als Kommentare enthalten. Sie legen eine Option fest, indem Sie das Zeichen # in der ersten Spalte löschen und den Wert in das richtige Verzeichnis ändern. In diesem Beispiel wird der Zugriff auf alle Objekte mit Ausnahme der Parameter ermöglicht:

   ```
   mozart directory = \\hub\mozart
   unison network directory = \\hub\unison
   # parameters directory = d:\maestro
   ```

3. Speichern Sie die Datei, und schließen Sie den Editor.
4. Damit die vorgenommenen Änderungen wirksam werden, müssen Tivoli Workload Scheduler und Netman gestoppt und erneut gestartet werden.

Wird eine Option nicht gesetzt bzw. ist diese nicht vorhanden, versucht das Composer-Programm von Tivoli Workload Scheduler, auf die Datenbankdateien auf dem lokalen Computer zuzugreifen.

Die lokalen Optionen auf dem Master einstellen

Wurden die Datenbankdateien aus den Standardverzeichnissen versetzt (wie unter 'Ein Verzeichnis gemeinsam benutzen' beschrieben), müssen Sie in den lokalen Optionen auf dem Hauptdomänenmanager die neue Position angeben. Siehe Abschnitt 'Die lokalen Optionen einstellen'.

Das Tivoli Workload Scheduler-Konto manuell erstellen

Bei Tivoli Workload Scheduler-Erstinstallationen wird ein Benutzerkonto mit bestimmten Berechtigungen benötigt, mit dem die Software installiert werden kann. Dieser Benutzer wird automatisch vom Setup-Programm erstellt. Soll das Konto manuell erstellt werden, führen Sie die unten aufgeführten Schritte durch.

Anmerkung: Für die folgenden Vorgehensweisen werden Kenntnisse über das Kontenmanagement unter Windows NT benötigt, einschließlich der Fähigkeit, Konten zu erstellen und Benutzerrechte zu vergeben. Verfügen Sie nicht über diese Kenntnisse, wenden Sie sich an Ihren NT-Administrator, oder lesen Sie die Microsoft-Dokumentation bzw. die Online-Hilfe zu Windows NT.

Tivoli Workload Scheduler-Verwaltungstypen

Die Datenbanken mit den Zeitplanungsobjekten von Tivoli Workload Scheduler befinden sich auf dem Computer, der als Hauptdomänenmanager definiert ist. Sie können die Zeitplanungsobjekte auf eine der folgenden Weisen verwalten:

1) Zentral, d. h. nur vom Hauptdomänenmanager aus

2) Dezentral, also von mehreren Tivoli Workload Scheduler-Computern aus

Wählen Sie eine der folgenden Prozeduren aus, je nachdem, auf welche Weise Sie die Zeitplanungsobjekte von Tivoli Workload Scheduler verwalten möchten.

Zentrale Verwaltung

Wenn Sie die Zeitplanungsobjekte nur vom Hauptdomänenmanager aus verwalten wollen, erstellen Sie den Benutzer (in der Regel *TWS* bzw. *maestro*) wie folgt:

1. Erstellen Sie ein lokales Benutzerkonto namens *TWS* auf dem Computer, der als Hauptdomänenmanager von Tivoli Workload Scheduler dienen soll. Beachten Sie, dass dieser Name auf allen Computern im Tivoli Workload Scheduler-Netzwerk verwendet werden muss, auch auf UNIX-Computern.

   Anmerkung: Sie können auch ein vorhandenes Benutzerkonto verwenden. Stellen Sie dabei aber sicher, dass dieser Benutzer zur Gruppe der NT-Administratoren gehört.

2. Ordnen Sie dem Benutzer *TWS* ein Basisverzeichnis zu. Beispiel:

   ```
   C:\win32app\maestro
   ```

   Geben Sie nicht das Stammverzeichnis eines Laufwerks an. Die Tivoli Workload Scheduler-Software wird im Basisverzeichnis dieses Benutzers installiert. In der Dokumentation wird dieses Verzeichnis als *TWShome* bezeichnet.

3. Erteilen Sie dem Benutzer *TWS* die folgenden erweiterten Benutzerrechte:

   Als Teil des Betriebssystems handeln
   Anheben einer Quote
   Anmelden als Stapelverarbeitungsauftrag
   Anmelden als Dienst
   Lokale Anmeldung
   Ersetzen eines Tokens auf Prozessebene

Sie können jetzt das Setup-Programm ausführen. Siehe „Die Software installieren” auf Seite 21.

Dezentrale Verwaltung

Wenn Sie die Zeitplanungsobjekte neben dem Hauptdomänenmanager von mehreren Tivoli Workload Scheduler-Computern aus verwalten wollen, erstellen Sie den Benutzer *TWS* mit Hilfe der folgenden Prozedur.

Anmerkung: Nach der Installation der Software müssen gemeinsam benutzte Verzeichnisse definiert werden, damit der Zugriff auf die Datenbanken des Hauptdomänenmanagers möglich ist. Dies wird unter „Die Verzeichnisse des Masters gemeinsam benutzen” auf Seite 29 beschrieben.

1. Erstellen Sie ein Domänenkonto namens *TWS* in der Domäne des Computers, der als Hauptdomänenmanager von Tivoli Workload Scheduler dienen soll. Beachten Sie, dass dieser Name innerhalb des Tivoli Workload Scheduler-Netzwerks auch auf UNIX-Computern verwendet werden muss.

   Anmerkung: Sie können auch ein vorhandenes Benutzerkonto verwenden. Stellen Sie dabei aber sicher, dass dieser Benutzer zur Gruppe der NT-Administratoren gehört.

2. Ordnen Sie dem Benutzer *TWS* ein Basisverzeichnis zu. Beispiel:

   ```
   C:\win32app\maestro
   ```

   Geben Sie nicht das Stammverzeichnis eines Laufwerks an. Die Tivoli Workload Scheduler-Software wird im Basisverzeichnis dieses Benutzers installiert. In der Dokumentation wird dieses Verzeichnis als *TWShome* bezeichnet.

3. Erteilen Sie dem Benutzer *TWS* auf jedem Computer, von dem aus Tivoli Workload Scheduler verwaltet wird, die folgenden erweiterten Benutzerrechte:

   Als Teil des Betriebssystems handeln
   Anheben einer Quote
   Anmelden als Stapelverarbeitungsauftrag
   Anmelden als Dienst
   Lokale Anmeldung
   Ersetzen eines Tokens auf Prozessebene

Sie können jetzt das Setup-Programm ausführen. Siehe „Die Software installieren” auf Seite 21.

Tivoli Workload Scheduler deinstallieren

Bei der Ausführung der Deinstallationsprogramme werden die Produktdateien, Registrierungsschlüssel, Dienste und Programmgruppen entfernt. Beachten Sie, dass bei der Deinstallation keine Dateien entfernt werden, die nach der Installation von Tivoli Workload Scheduler erstellt wurden, bzw. Dateien, die zum Zeitpunkt der Deinstallation geöffnet sind.

Sie müssen das Windows NT-Tool **Software** verwenden, um Tivoli Workload Scheduler zu deinstallieren.

Das Tool ″Software″ in der Systemsteuerung von Windows NT verwenden

Tivoli Workload Scheduler kann unter Windows NT nur vom Benutzer *TWS* deinstalliert werden, der über Administratorrechte unter Windows NT verfügt.

Gehen Sie wie folgt vor:

1. Stoppen Sie Tivoli Workload Scheduler und alle Tivoli Workload Scheduler-Dienste. Nähere Informationen hierzu finden Sie unter „Tivoli Workload Scheduler trennen und stoppen” auf Seite 20.
2. Klicken Sie **Software** im Ordner **Systemsteuerung** an.
3. Wählen Sie die Indexzunge **Installieren/Deinstallieren** aus, und klicken Sie **Tivoli Workload Scheduler (Benutzer=***benutzername***)** an, um das Programm zu entfernen.
4. Klicken Sie **Hinzufügen/Entfernen...** an. Klicken Sie im daraufhin angezeigten Dialogfenster **Ja** an.
5. Verlassen Sie das Fenster **Eigenschaften von Software**.
6. Fahren Sie das System herunter, und starten Sie den Computer erneut.

Die Dateien, die bei der Installation des Produkts erstellt wurden, werden bei der Deinstallation entfernt. Die Dateien, die nach der Installation erstellt oder geändert wurden, werden nicht entfernt. Diese Dateien können Sie bei einer späteren Installation von Tivoli Workload Scheduler verwenden.

3

Unter UNIX installieren

In diesem Abschnitt wird der Installationsprozess von Tivoli Workload Scheduler für UNIX-Systeme und -Netzwerke beschrieben.

Systemvoraussetzungen

Nachfolgend sind die Systemvoraussetzungen für Tivoli Workload Scheduler auf UNIX-Systemen aufgeführt:

- CD-ROM-Laufwerk für die Installation
- Ca. 120 MB freier Plattenspeicherplatz für Domänenmanager und fehlertolerante Agenten. Ca. 80 MB für Standardagenten.
- Tivoli Workload Scheduler erstellt außerdem Protokolldateien und temporäre Dateien, die auf dem lokalen Festplattenlaufwerk gespeichert werden. Der benötigte Plattenspeicherplatz hängt von der Anzahl der von Tivoli Workload Scheduler verwalteten Jobs ab sowie davon, wie lange die Protokolldateien gespeichert bleiben sollen.
- 32 MB Arbeitsspeicher und 48 MB Auslagerungsspeicher für Domänenmanager und fehlertolerante Agenten. Der Bedarf von Standardagenten ist geringer.
- Bei fehlertoleranten Agenten können optional auch Verzeichnisse vom Hauptdomänenmanager angehängt werden. Dafür ist das Network File System (NFS) erforderlich. Ziehen Sie Ihre System- und Netzwerkadministratoren zu Rate, und schlagen Sie weitere Informationen im *Tivoli Workload Scheduler Console Benutzerhandbuch* nach.
- Unter Digital UNIX muss die Umgebungsvariable LD_NO_LIBSTREAMSOCKET auf 0 gesetzt und vor der Installation von Tivoli Workload Scheduler exportiert werden.

Lesen Sie die *Tivoli Workload Scheduler Release-Informationen,* bevor Sie fortfahren.

Für Endpunkt-zu-Endpunkt-Zeitplanung installieren

Wenn Sie Tivoli Workload Scheduler auf einer Workstation installieren, die als ein verteilter Agent für die Endpunkt-zu-Endpunkt-Zeitplanung (ein Domänenmanager, ein fehlertoleranter Agent oder ein Standardagent) verwendet wird, müssen Sie Folgendes ausführen:

1. Geben Sie OPCMASTER als Name des Hauptdomänenmanagers im Skript *customize* an.
2. Definieren Sie nach der Installation die WEZ-Zeitzone für die Workstation. Dies ist notwendig, damit Start- und Endzeiten auf dem Tivoli Workload Scheduler for z/OS-Controller mit der lokalen Controller-Zeitzone gemeldet und die Zeitabhängigkeiten richtig aufgelöst werden können. Fügen Sie die folgenden Zeilen dem Shell-Skript **StartUp** hinzu:

   ```
   TZ=GMT
   ```

```
export TZ
```

Näheres hierzu finden Sie unter „Zeitzonen” auf Seite 89.

Installationsverfahren

Befolgen Sie diese Anweisungen für UNIX-Computer, wenn Tivoli Workload Scheduler zum ersten Mal installiert wird bzw. wenn Tivoli Workload Scheduler vollständig deinstalliert wurde. Informationen zu Computern mit Windows NT finden Sie unter Kapitel 2, „Unter Windows NT installieren” auf Seite 19.

Die Tivoli Workload Scheduler-Steuerkomponente installieren

Führen Sie folgende Schritte durch, um Tivoli Workload Scheduler auf einem UNIX-Computer zu installieren:

1. Erstellen Sie den Tivoli Workload Scheduler-Benutzer. Die Software wird im Basisverzeichnis des Benutzers installiert, auf das mit *twshome* verwiesen wird.

   Benutzer:
   maestro

   Gruppe:
   tivoli

   Basisverzeichnis:
   twshome (z. B.: /opt/maestro)

   Anmerkung: Wenn Sie mehr als eine Instanz von Tivoli Workload Scheduler auf demselben Computer installieren, sollten Sie berücksichtigen, dass bei jeder Installation versucht wird, die Dateien in das Verzeichnis *twshome*/../unison zu kopieren. Geben Sie unterschiedliche übergeordnete Verzeichnisse an, um Konflikte zu vermeiden. Beispiel: **/opt/maestro** und **/opt/maestro2**.

2. Legen Sie das Installationsband oder die Installations-CD ein, und schreiben Sie die Software zurück.

   a. Melden Sie sich als **root** an, und wechseln Sie in das Verzeichnis *twshome*.

   b. Extrahieren Sie die Software:

   ```
   tar -xvf cd/TWS/plattform/MAESTRO.TAR
   ```

   Dabei gilt Folgendes:

   cd Der Pfadname Ihres CD-ROM-Laufwerks

   plattform
   Der Plattformtyp. Es gibt folgende Möglichkeiten:

   AIX (bei IBM)

   DYNIX (bei IBM-Sequent Numa)

   HPUX (bei Hewlett Packard UNIX)

   IRIX (bei SGI Irix)

   LINUX

   - **I386** (bei Linux auf Intel)
   - **S390** (bei Linux auf 390)

(bei Data General UNIX)

OSF (bei Compaq True64)

SOLARIS (bei Sun Solaris)

3. Führen Sie das Skript **customize** aus.

   Ein Beispielskript **customize** für die Workstation eines Hauptdomänenmanagers könnte folgendermaßen lauten:

   ```
   /bin/sh customize -new -thiscpu mdm -master mdm -uname twsuser -group gruppe1 [andere_opt]
   ```

   Ein Beispielskript **customize** für eine FTA-Workstation könnte folgendermaßen lauten:

   ```
   /bin/sh customize -new -thiscpu dm1 -master mdm -uname twsuser -group gruppe1 [optionen]
   ```

   Wenn Sie ein HP-UX 10.x-System verwenden, befindet sich Ihre Bourne-Shell unter Umständen im Verzeichnis **/usr**. In diesem Fall verwenden Sie folgende Syntax:

   ```
   /usr/bin/sh customize -new -thiscpu mdm -master mdm -uname twsuser -group gruppe1 [andere_opt]
   ```

   Weitere Informationen zu den Argumenten von **customize** und weitere Beispiele finden Sie unter Das Skript ″customize″ auf Seite 43.

4. Erstellen Sie eine Datei des Typs .profile für den Maestro-Benutzer, wenn noch keine vorhanden ist (*twshome*/.profile). Fügen Sie in dieser Datei der PATH-Variablen *twshome* und *twshome*/**bin** hinzu. Beispiel:

   ```
   PATH=/bin:/usr/bin:/usr/local/bin:/opt/maestro:/opt/maestro/bin:$PATH
   ```

   Endet die PATH-Anweisung mit einem Punkt (.), sollten Sie den Punkt durch die oben genannten Pfadangaben ersetzen bzw. die Pfadangaben vor dem Punkt einfügen. Enthält die PATH-Anweisung keinen Punkt, fügen Sie die Pfadangaben am Ende hinzu.

5. Wenn der Tivoli Workload Scheduler-Netzwerkmanagementprozess Netman bei jedem Bootvorgang des Systems als Dämon gestartet werden soll, fügen Sie der Datei **/etc/rc** oder der entsprechenden Datei in Ihrem System eine der folgenden Sequenzen hinzu.

   Wenn nur Netman gestartet werden soll:

   ```
   if [-x twshome/StartUp]
   then
   echo "Netman wird gestartet..."
   /bin/su - maestro -c " twshome/StartUp"
   fi
   ```

   Wenn der gesamte Tivoli Workload Scheduler-Prozessbaum gestartet werden soll:

   ```
   if [-x twshome/bin/conman]
   then
   echo "Workload Scheduler wird gestartet..."
   /bin/su - maestro -c " twshome/bin/conman start"
   fi
   ```

Der Tivoli Workload Scheduler-Installationsprozess ist jetzt abgeschlossen.

Konfigurationsschritte

Führen Sie bei UNIX-Installationen die nachfolgenden Konfigurationstasks aus. Diese Tasks müssen von den Befehlszeilenschnittstellen **Composer** und **Conman** aus ausgeführt werden.

Wenn Sie auf diese Weise den Hauptdomänenmanager konfiguriert haben, können Sie die Job Scheduling Console zur Konfiguration der anderen Workstations und Job Scheduling-Objekte in Ihrem Tivoli Workload Scheduler-Netzwerk verwenden. Weitere Informationen zu den unten genannten Befehlen finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*. Im Handbuch *Tivoli Workload Scheduler Console Benutzerhandbuch* finden Sie weitere Informationen zur Verwendung der Job Scheduling Console, um andere Workstations im Netzwerk zu konfigurieren.

Einen Hauptdomänenmanager nach der Installation konfigurieren

1. Melden Sie sich beim Hauptdomänenmanager als **maestro** an. Dies ist die ID-Variable *tws_user*, die unter UNIX standardmäßig den Wert **maestro** annimmt. Die UNIX-ID *tws_user* kann mit dem Argument **-uname** des Skripts **customize** geändert werden.
2. Erstellen Sie die Workstationdefinition für einen Hauptdomänenmanager in der Tivoli Workload Scheduler-Datenbank mit Hilfe der Composer-Befehlszeile. Öffnen Sie ein Befehlszeilenfenster in UNIX, und geben Sie folgende Befehle ein:

   ```
   composer
   new
   ```

3. Hiermit wird ein Texteditor aufgerufen, in dem Sie die Workstationdefinition für einen Hauptdomänenmanager in der Tivoli Workload Scheduler-Datenbank erstellen können. Nachfolgend finden Sie ein Beispiel einer Workstationdefinition für einen Hauptdomänenmanager. Weitere Informationen zu Workstationdefinitionen finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

   ```
   cpuname MDM
   os UNIX
   node master.santaclara.tivoli.com
   description "Hauptdomänenmanager"
   for Maestro
   autolink on
   resolvedep on
   fullstatus on
   end
   ```

4. Erstellen Sie die Definition für das Jobnetz **final**. Dieses Jobnetz enthält den Job **Jnextday**, der die Erstellung der Symphony-Datei automatisiert.

   ```
   composer "add Sfinal"
   ```

   Nach der Installation muss der Schritt einmal manuell ausgeführt werden, danach kann er automatisiert werden. Näheres finden Sie unter „Den Produktionszyklus automatisieren” auf Seite 83.

5. Führen Sie den Job **Jnextday** aus:

   ```
   Jnextday
   ```

   Nach der Installation muss der Schritt einmal manuell ausgeführt werden, danach kann er automatisiert werden. Näheres finden Sie unter „Den Produktionszyklus automatisieren” auf Seite 83.

6. Überprüfen Sie den Status von Tivoli Workload Scheduler, nachdem der Job **Jnextday** beendet wurde:

   ```
   conman status
   ```

   Wenn Tivoli Workload Scheduler korrekt gestartet wurde, ist der Status **Batchman=LIVES**.

7. Erhöhen Sie die Begrenzung für die Ausführung von Jobs. Die Standardjobbegrenzung nach der Installation ist auf null gesetzt, d. h., es können keine Jobs ausgeführt werden; daher sollten Sie die Jobbegrenzung jetzt folgendermaßen erhöhen:

   ```
   conman "limit;10"
   ```

Sie können jetzt beginnen, zusätzliche Zeitplanungsobjekte in der CLI zu definieren, wie Workstations, Jobs und Jobnetze, oder Sie können mit der Installation der Job Scheduling Console und des Connectors fortfahren. Beachten Sie, dass die neuen Objekte erst dann erkannt werden, wenn der Job **Jnextday** im Jobnetz **final** ausgeführt wird. Standardmäßig wird das Jobnetz **final** um 5:59 Uhr ausgeführt (diese Standardeinstellung können Sie ändern; siehe „Die Tagesstartzeit ändern” auf Seite 85). Wenn Sie die neuen Zeitplanungsobjekte zu einem früheren Zeitpunkt aufnehmen möchten, können Sie `conman "release final"` ausführen oder den Conman-Befehl *submit* verwenden. Informationen zur Definition der Zeitplanungsobjekte finden Sie in den Handbüchern *Tivoli Workload Scheduler Referenz* oder *Tivoli Workload Scheduler Console Benutzerhandbuch*.

Einen fehlertoleranten Agenten nach der Installation konfigurieren

1. Melden Sie sich auf der FTA-Workstation als **maestro** an. Dies ist die ID-Variable *tws_user*, die unter UNIX standardmäßig den Wert **maestro** annimmt. Die UNIX-ID *tws_user* kann mit dem Argument **-uname** des Skripts **customize** geändert werden.
2. Erstellen Sie die FTA-Workstationdefinition in der Tivoli Workload Scheduler-Datenbank mit Hilfe der Composer-Befehlszeile. Öffnen Sie ein Befehlszeilenfenster in UNIX, und geben Sie folgende Befehle ein:

   ```
   composer
   new
   ```

3. Hiermit wird ein Texteditor aufgerufen, in dem Sie die FTA-Workstationdefinition in der Tivoli Workload Scheduler-Datenbank erstellen können. Nachfolgend finden Sie ein Beispiel einer Workstationdefinition für einen FTA. Weitere Informationen zu Workstationdefinitionen finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

   ```
   cpuname DM1
   os UNIX
   node domain1.santaclara.tivoli.com
   description "Domänenmanager"
   for Maestro
   autolink on
   end
   ```

4. Führen Sie den Befehl **StartUp** aus, um Netman zu starten.

   ```
   conman StartUp
   ```

5. Erstellen Sie eine neue Symphony-Datei, die die FTA-Workstationdefinition enthält. Führen Sie hierzu den Job **Jnextday** auf dem Hauptdomänenmanager aus, wodurch die Erstellung einer neuen Symphony-Datei automatisiert wird:

   ```
   Jnextday
   ```

6. Führen Sie den Befehl 'link' vom Hauptdomänenmanager aus durch, um eine Verbindung zum FTA herzustellen und die Symphony-Datei darauf herunterzuladen:

   ```
   conman "link FTA-name"
   ```

Tivoli Workload Scheduler aktualisieren

Aktualisieren Sie mit der nachfolgend beschriebenen Prozedur vorhandene Installationen von Tivoli Workload Scheduler auf UNIX-Computern. Es wird empfohlen, bei der Aktualisierung eines Tivoli Workload Scheduler-Netzwerks zunächst alle Agentenworkstations und anschließend den Hauptdomänenmanager zu aktualisieren.

Anmerkung: Lesen Sie die *Tivoli Workload Scheduler Release-Informationen*, da diese zusätzliche Informationen zum Aktualisieren vorhandener Software enthalten.

Die nachfolgenden Schritte zur Durchführung einer Aktualisierung werden in der Reihenfolge beschrieben, in der sie auch ausgeführt werden sollten.

Vorbereitung: Tivoli Workload Scheduler stoppen

1. Unterbrechen Sie über die Job Scheduling Console die Verbindung der Zielworkstation mit den anderen Workstations im Netzwerk. Sie können auch an der Befehlszeile den folgenden Befehl eingeben:

   ```
   conman "unlink workstationname;noask"
   ```

2. Stoppen Sie die Zielworkstation über die Job Scheduling Console, und fahren Sie sie herunter. Geben Sie an der Befehlszeile die folgenden Befehle ein:

   ```
   conman "stop"
   conman "shut;wait"
   ```

3. Wenn Sie die Aktualisierung auf einem Agenten durchführen, hängen Sie die angehängten NFS-Verzeichnisse auf dem Hauptdomänenmanager ab (umount).

Sicherungskopien

Das Skript **customize** versetzt Ihre Arbeitskopien von **Sfinal**, **Jnextday**, **NetConf** und **StartUp** in ein Verzeichnis mit dem Namen *twshome*/**config.old**. Diese Dateien werden häufig angepasst, um den besonderen Anforderungen des Benutzers zu entsprechen; Sie können mit den gesicherten Kopien nach der Aktualisierung Ihre Änderungen an den Dateien übernehmen. Das Skript **customize** überschreibt keine Dateien in den Verzeichnissen **mozart**, **stdlist** oder **unison**, die nach der Installation von Tivoli Workload Scheduler geändert wurden.

Wenn Sie andere Dateien während des Aktualisierungsprozesses schützen möchten, sollten Sie diese jetzt kopieren oder umbenennen. Vorsorglich sollten Sie eine Sicherungskopie der folgenden Verzeichnisse bzw. Datei erstellen:

- Das Verzeichnis *twshome*.
- Das Netman-Basisverzeichnis (*twshome/../Unison)*
- Die Komponentendatei (normalerweise */usr/unison/components)*

Die Software aktualisieren und ″customize″ ausführen

1. Legen Sie das Installationsband oder die Installations-CD ein, und schreiben Sie die Software zurück.
   a. Melden Sie sich als **root** an, und wechseln Sie in das Verzeichnis *twshome*.
   b. Extrahieren Sie die Software:
      `tar xvf cd/TIVOLI/plattform/MAESTRO.TAR`

      Dabei gilt Folgendes:

      cd Der Pfadname Ihres CD-ROM-Laufwerks

      plattform
      Der Plattformtyp. Es gibt folgende Möglichkeiten:

      AIX (bei IBM)

      DYNIX (bei IBM-Sequent Numa)

      HPUX (bei Hewlett Packard UNIX)

      IRIX (bei SGI Irix)

      LINUX

      - **I386** (bei Linux auf Intel)
      - **S390** (bei Linux auf 390)

      (bei Data General UNIX)

      OSF (bei Compaq True64)

      SOLARIS (bei Sun Solaris)

2. Führen Sie das Skript **customize** aus.
   Ein Beispielskript **customize** für die Workstation eines Hauptdomänenmanagers könnte folgendermaßen lauten:
   `/bin/sh customize -update [-uname name] [andere_opt]`

   Ein Beispielskript **customize** für eine FTA-Workstation könnte folgendermaßen lauten:
   `/bin/sh customize -update [-uname name] [optionen]`

   Wenn Sie ein HP-UX 10.x-System verwenden, befindet sich Ihre Bourne-Shell unter Umständen im Verzeichnis **/usr**. In diesem Fall verwenden Sie folgende Syntax:
   `/usr/bin/sh customize -update [-uname name] [andere_opt]`

   Weitere Informationen zu den Argumenten von **customize** und weitere Beispiele finden Sie unter Das Skript ″customize″ auf Seite 43.

Das Sicherheitsprofil aktualisieren

Bei der Migration von Maestro Version 6.0 muss eine zusätzliche Maestro-Benutzerdefinition der Sicherheitsdatei hinzugefügt werden. Die ID *tme_region_administrator* muss in die Sicherheitsdatei aufgenommen werden, wenn Sie Tivoli Management Framework für die Verwendung des Tivoli Workload Scheduler-Connectors bereits installiert haben.

1. Melden Sie sich als **Maestro** an.
2. Führen Sie den Befehl **dumpsec** aus, um eine editierbare temporäre Kopie der Sicherheitsdatei zu erstellen:

   ```
   dumpsec > tempsec
   ```

3. Editieren Sie die Sicherheitsdatei. Fügen Sie im Abschnitt USER MAESTRO die Tivoli-Administrator-ID hinzu:

   ```
   USER MAESTRO
   root, maestro, administrator, tme_region_admin
   ```

4. Führen Sie den Befehl **makesec** aus, um die temporäre Datei in eine neue Sicherheitsdatei zu kompilieren:

   ```
   makesec tempsec
   ```

   Weitere Informationen zu den Befehlen **Makesec** und **Dumpsec** finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

Ihre Dateien wiederherstellen

Verwenden Sie Ihre zuvor zu Referenzzwecken gesicherten Dateien, um die neuen Versionen der Dateien entsprechend zu ändern.

Tivoli Workload Scheduler erneut starten

So starten Sie Tivoli Workload Scheduler erneut:

1. Melden Sie sich mit der Benutzer-ID **TWS** oder **maestro** an.
2. Führen Sie den Befehl *start* aus:

   ```
   conman start
   ```

3. Führen Sie den Befehl *link* aus:
   - Bei einem Hauptdomänenmanager:

     ```
     conman "link @"
     ```

   - Bei einer FTA-Workstation:

     ```
     conman "link domänenname"
     ```

Die Softwareaktualisierung auf dieser Workstation ist damit abgeschlossen.

Das Skript ″customize″

Verwenden Sie das Skript **customize**, um Tivoli Workload Scheduler auf UNIX-Plattformen zu installieren und zu aktualisieren.

Syntax

customize -new -thiscpu *workstationname* **-master** *workstationname* [**-company** *″ firmenname″*] [**-group** *gruppenname*] [**-nmhome** *netman-basisverzeichnis*] [**-noexp**] [**-nolinks**| **-execpath** *pfadname*] [**-uname** *benutzername*][**-port** *netman-anschluss*]

customize -update [**-company** *″ firmenname″*] [**-group** *gruppenname*] [**-uname** *benutzername*]

Beschreibung

Mit dem Skript **customize** wird Tivoli Workload Scheduler installiert bzw. aktualisiert. Sie können folgende Funktionen durchführen:

- Erstinstallation von Tivoli Workload Scheduler: Tivoli Workload Scheduler und Netman installieren. Eine Komponentendatei mit neuen Einträgen erstellen.
- Tivoli Workload Scheduler-Aktualisierungen: Tivoli Workload Scheduler und Netman gegebenenfalls aktualisieren. Einträge in der Komponentendatei aktualisieren. Berechtigungen auf ihre Standardwerte zurücksetzen, vorausgesetzt, die Originaldatei MAESTRO.TAR befindet sich nicht im Verzeichnis *twshome*.
- Einzelheiten des Installationsprozesses werden in einer Datei namens **customize.log** protokolliert. Diese Datei befindet sich im selben Verzeichnis, von dem aus Sie das Skript **customize** ausführen.

Argumente

-new Bei einer Erstinstallation.

-update
Bei der Aktualisierung einer vorhandenen Installation. Beachten Sie, dass sich der von Tivoli Workload Scheduler verwendete Datenbanktyp bei einer Softwareaktualisierung nicht ändert.

-thiscpu
Der Name dieser Workstation. Bei der nicht erweiterten Version von Tivoli Workload Scheduler kann der Name aus bis zu acht alphanumerischen Zeichen, Bindestrichen (-) oder Unterstreichungszeichen (_) bestehen, wobei das erste Zeichen ein Buchstabe sein muss. Bei der erweiterten Version darf er bis zu sechzehn Zeichen umfassen. Mit diesem Namen wird zu einem späteren Zeitpunkt die Workstation in Tivoli Workload Scheduler formal definiert.

-master
Der Name des Hauptdomänenmanagers. Bei der nicht erweiterten Version von Tivoli Workload Scheduler kann der Name aus bis zu acht alphanumerischen Zeichen, Bindestrichen (-) oder Unterstreichungszeichen (_) bestehen, wobei das erste Zeichen ein Buchstabe sein muss. Bei der erweiterten Version darf er bis zu sechzehn Zeichen umfassen. Ist dieser Computer der Hauptdomänenmanager oder eine eigenständige Workstation, geben Sie den Namen von **-thiscpu** ein. Mit diesem Namen wird zu einem späteren Zeitpunkt die Workstation in Tivoli Workload Scheduler formal definiert.

-company
Der Name des Unternehmens in doppelten Anführungszeichen (kann aus bis zu 40 Zeichen bestehen). Der Name erscheint in Programm-Headern und Berichten.

-group
Der Name der Produktgruppe, in der Tivoli Workload Scheduler installiert bzw. aktualisiert werden soll. Der Name kann aus maximal 40 Zeichen bestehen, beim ersten Zeichen muss es sich um einen Buchstaben handeln. Wird diese Einstellung bei einer Erstinstallation oder einer ersten Aktualisierung einer früheren Version von Tivoli Workload Scheduler übergangen, wird der Name auf den Standardwert **DEFAULT** gesetzt. Wird der Name bei einer gewöhnlichen Aktualisierung (also nicht der ersten Aktualisierung einer früheren Version von Tivoli Workload Scheduler) angegeben, wird er mit dem Gruppennamen aus der Komponentendatei überschrieben; das Basisverzeichnis des Benutzers wird dabei als Referenz verwendet. Weitere Informationen finden Sie unter „Produktgruppen” auf Seite 16.

-nmhome
Das Netman-Basisverzeichnis. Wird es nicht angegeben, wird Netman im Basisverzeichnis von Tivoli Workload Scheduler installiert (*twshome*). Weitere Informationen hierzu finden Sie unter „Basisverzeichnis von Netman” auf Seite 17. Diese Option wirkt sich nicht auf Aktualisierungen aus.

-noexp
Es werden nicht erweiterte Datenbanken verwendet. Die globale Option *expanded version* wird auf *no* gesetzt; es werden nicht erweiterte Datenbanken erstellt. Da frühere Versionen keine erweiterten Datenbanken unterstützen, sollten Sie dieses Schlüsselwort verwenden, wenn Sie Tivoli Workload Scheduler in einem Netzwerk installieren, in dem sich Computer mit einer beliebigen Version von Maestro for MPE befinden. Auf allen Computern mit Maestro Version 6.0 oder höher oder Tivoli Workload Scheduler können die Datenbanken durch Ausführen des Befehls **dbexpand** auf dem Hauptdomänenmanager erweitert werden. Weitere Informationen zum Befehl **dbexpand** finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*. Wird diese Option nicht angegeben, wird die globale Option ’expanded version’ auf ’yes’ gesetzt; somit werden erweiterte Datenbanken erstellt.

[**-nolinks**|**-execpath** *pfadname*]
Mit der Option zum Verbinden (link) wird der Pfad festgelegt, den das Skript **customize** zur Erstellung von Verbindungen zu den Tivoli Workload Scheduler-Dienstprogrammbefehlen verwendet. Wenn Sie die Option ’-nolinks’ angeben, werden keine Verbindungen erstellt. Wenn Sie die Option ’-execpath’ angeben, werden basierend auf dem angegebenen Pfad Verbindungen erstellt. Wenn die Verbindungsoption ganz weggelassen wird, werden folgende Verbindungen erstellt:

usr/bin/mat	*twshome/bin/at*
usr/bin/mbatch	*twshome/bin/batch*
usr/bin/datecalc	*twshome/bin/datecalc*
usr/bin/jobstdl	*twshome/bin/jobstdl*
usr/bin/maestro	*twshome/bin/maestro*
usr/bin/mdemon	*twshome/bin/mdemon*
usr/bin/morestdl	*twshome/bin/morestdl*
usr/bin/muser	*twshome/bin/muser*
usr/bin/parms	*twshome/bin/parms*

-uname

Der Name des Benutzers, für den Tivoli Workload Scheduler installiert bzw. aktualisiert wird. Der Name darf keinen Punkt (.) enthalten, da solche Namen für MPE-Anmeldenamen reserviert sind. Die Software wird im Basisverzeichnis dieses Benutzers installiert bzw. aktualisiert. Wird diese Option nicht angegeben, wird der Standardbenutzername *maestro* verwendet.

-port Die TCP-Anschlussnummer, auf die Netman auf dem lokalen Computer reagiert. Dabei muss es sich um einen 16-Bit-Wert ohne Vorzeichen im Bereich von 1 bis 65535 handeln (Beachten Sie, dass die Werte zwischen 0 und 1023 für Standarddienste, wie z. B. FTP, TELNET, HTTP usw., reserviert sind.). Der Standardwert ist 31111. Sie können diesen Wert jederzeit in der Datei mit den lokalen Optionen ändern.

Tivoli Workload Scheduler deinstallieren

Gehen Sie wie folgt vor, um Tivoli Workload Scheduler von einem UNIX-System zu deinstallieren:

1. Stoppen Sie vor der Deinstallation alle vorhandenen Tivoli Workload Scheduler-Prozesse, die auf diesem speziellen UNIX-System erstellt wurden.
2. Melden Sie sich auf dem UNIX-System als Root-Benutzer an.
3. Erstellen Sie von allen Tivoli Workload Scheduler-Datenbankdateien, die Sie weiterverwenden wollen, eine Sicherungskopie. Die Datenbankdateien sind im Basisverzeichnis des Tivoli Workload Scheduler-Benutzers gespeichert und lauten wie folgt:
   - /parameters
   - /parameters.KEY
   - /mozart/calendars
   - /mozart/calendars.KEY
   - /mozart/jobs
   - /mozart/jobs.KEY
   - /mozart/mastsked
   - /mozart/mastsked.KEY
   - /mozart/prompts
   - /mozart/prompts.KEY
   - /mozart/resources
   - /mozart/resources.KEY
   - /../unison/network/cpudata
   - /../unison/network/cpudata.KEY
   - /../unison/network/userdata
   - /../unison/network/userdata.KEY

   Anmerkung: Die Datenbank **job.sched** wird automatisch wiederhergestellt, Sie brauchen sie also nicht zu speichern.
4. Überprüfen Sie den Inhalt der Datei */usr/unison/components*. Wenn die Datei mehrere Einträge enthält, die mit verschiedenen Maestro/Tivoli Workload Scheduler-Konten (Produktgruppen) übereinstimmen, löschen Sie aus der Datei die Zeilen, die mit der Instanz übereinstimmen, die Sie entfernen wollen.

   Beispielsweise könnte */usr/unison/components* folgende Einträge enthalten:

   ```
   netman  1.5 /opt/maestro  DEFAULT
   maestro 7.0 /opt/maestro  DEFAULT
   netman  1.3 /home/maestro dev_test
   maestro 6.1 /home/maestro dev_test
   ```

   Wenn Sie vorhaben, die Tivoli Workload Scheduler-Instanz unter */home/maestro* zu entfernen, löschen Sie die beiden letzten Zeilen.

   Enthält */usr/unison/components* nur die Instanz, die Sie entfernen wollen, löschen Sie die gesamte Datei.

5. Entfernen Sie gegebenenfalls die Verbindungen zum Verzeichnis */usr/bin*. Der Installationsprozess bietet Ihnen die Option, ausführbare Dateien von Tivoli Workload Scheduler mit einem gemeinsam benutzten Verzeichnis zu verbinden. */usr/bin* ist das Standardverzeichnis. Entfernen Sie folgende Dateien:
   - /usr/bin/maestro
   - /usr/bin/mat
   - /usr/bin/mbatch
   - /usr/bin/datecalc
   - /usr/bin/morestdl
   - /usr/bin/jobstdl
   - /usr/bin/parms
6. Entfernen Sie zum Schluss mit den folgenden Befehlen das gesamte Maestro/ Tivoli Workload Scheduler-Konto:

   ```
   rm -rf <twshome>/../unison
   rm -rf <twshome>
   ```

   Wenn dem Systemstartbefehl der Befehl *conman "start"* oder *<twshome>/StartUp* hinzugefügt wurde, müssen Sie auch diese Einträge entfernen.

4

Sicherheit definieren

Die Tivoli Workload Scheduler-Programme und -Befehle ermitteln die Berechtigungen der einzelnen Benutzer durch einen Vergleich des Benutzernamens mit den Benutzerdefinitionen in der Sicherheitsdatei (**Security**). In diesem Kapitel erfahren Sie, wie Benutzerdefinitionen geschrieben werden und die Sicherheitsdatei verwaltet wird.

Die Datei "Security"

Mit der Software wird eine Schablonendatei namens *TWShome*/**config/Security** zur Verfügung gestellt. Während der Installation wird eine Kopie der Schablone unter *TWShome*/**Security** installiert; eine kompilierte, lauffähige Kopie wird unter *TWShome*/**../unison/Security** installiert.

Die Datei "Security" erstellen

Editieren Sie die Schablonendatei *TWShome*/**Security**, um Benutzerdefinitionen zu erstellen. Die ursprüngliche Schablone im Verzeichnis *TWShome*/**config/Security** darf nicht geändert werden. Verwenden Sie dann den Befehl **makesec**, um eine neue aktive Sicherheitsdatei zu kompilieren und zu installieren. Nachdem diese installiert ist, können Sie weitere Änderungen vornehmen, indem Sie eine editierbare Kopie der aktiven Datei mit dem Befehl **dumpsec** erstellen. Die Befehle **makesec** und **dumpsec** werden in diesem Kapitel weiter unten beschrieben. Die Änderungen an der Sicherheitsdatei werden erst wirksam, nachdem Sie Tivoli Workload Scheduler gestoppt und erneut gestartet haben.

Sicherheit in einem Netzwerk verwalten

Jede Workstation in einem Tivoli Workload Scheduler-Netzwerk (Domänenmanager, fehlertolerante Agenten und Standardagenten) verfügt über ihre eigene Sicherheitsdatei. Sie können entweder auf jeder Workstation eine eigene Datei verwalten oder aber auf dem Hauptdomänenmanager eine Sicherheitsdatei erstellen und diese anschließend auf alle Domänenmanager, fehlertoleranten Agenten und Standardagenten kopieren.

Anmerkung: Vergessen Sie nicht, in der Sicherheitsdatei nach der Installation von Tivoli Management Framework und des Tivoli Workload Scheduler-Connectors den Tivoli-Administrator hinzuzufügen. Näheres hierzu finden Sie im Kapitel zur Installation des Tivoli Workload Scheduler-Connector im Handbuch *Tivoli Workload Scheduler Console Benutzerhandbuch*.

Sicherheit für den Tivoli-Administrator definieren

Damit Sie die Job Scheduling Console auf einem Master oder einem fehlertoleranten Agenten verwenden können, muss/müssen der/die Tivoli-Administrator(en) in der Sicherheitsdatei des Masters oder fehlertoleranten Agenten definiert sein. Ein Benutzereintrag muss mit folgender Zeichenfolge definiert sein:

```
CPU=$FRAMEWORK
```

| und

| `LOGON="tme-admin"`

| Dabei kann es sich um einen separaten Benutzereintrag oder einen bereits vorhandenen
| Benutzereintrag handeln. Siehe die Beispiele unter „Beispiel der Datei ″Security″” auf Seite 60.

Syntax der Datei ″Security″

Die Sicherheitsdatei enthält mindestens eine Benutzerdefinition. Jede Benutzerdefinition kennzeichnet eine Gruppe von Benutzern sowie die Objekte und Vorgänge, auf die sie zugreifen bzw. die sie ausführen können.

Benutzerdefinitionen

Eine Benutzerdefinition kennzeichnet eine Gruppe von Benutzern sowie die Objekte und Aktionen, auf die sie zugreifen bzw. die sie ausführen können.

Syntax

[**#** *kommentar*]

user *def-name benutzerattribute*

begin [***** *kommentar*]

objektart [*objektattribute*] **access**[=*aktion*[**,**...]]

[*objektart ...*]

[**end**]

Variablen

[# | ***]** *kommentar*
Sämtlicher Text nach einem Nummernzeichen oder Stern und mindestens einem Leerzeichen wird als Kommentar behandelt. Die Kommentare werden nicht in die aktive Sicherheitsdatei kopiert, die mit dem Befehl **makesec** installiert wird.

def-name
Gibt den Namen der Benutzerdefinition an. Der Name kann bis zu 36 alphanumerische Zeichen enthalten und muss mit einem Buchstaben beginnen.

benutzerattribute
Gibt mindestens ein Attribut an, mit dem die Gruppe von Benutzern festgelegt wird, für die diese Definition gilt. Die Benutzerattribute werden wie folgt angegeben:

benutzerattribut[{+ | ~}*benutzerattribut*[...]]

Mit einem Pluszeichen (+) werden die Attribute hinzugefügt, über die die Benutzer verfügen müssen. Mit einem Tildezeichen (~) werden die Attribute hinzugefügt, über die die Benutzer nicht verfügen dürfen. Ein Benutzerattribut wird wie folgt definiert:

cpu=*workstation*|**$framework**|**@ [,...]**
Dabei gilt Folgendes:

workstation
| Gibt die Workstation an, an der der Benutzer angemeldet ist.
| Platzhalterzeichen sind erlaubt. Folgende Tivoli Workload Scheduler-
| Variablen können verwendet werden:

$master
Der Benutzer ist am Tivoli Workload Scheduler-Hauptdomänenmanager angemeldet.

$remotes
Der Benutzer ist an einem Tivoli Workload Scheduler-Standardagenten angemeldet.

$slaves
Der Benutzer ist an einem fehlertoleranten Agenten von Tivoli Workload Scheduler angemeldet.

$thiscpu
Der Benutzer ist an der Tivoli Workload Scheduler-Workstation angemeldet, auf der das gesicherte Programm ausgeführt wird.

Verwenden Sie für Job Scheduling Console-Benutzer **$framework**.

$framework
Gibt die Workstation an, von der aus der Benutzer die Job Scheduling Console ausführt.

@ Gibt an, dass der Benutzer über die Job Scheduling Console auf Tivoli Workload Scheduler zugreift oder dass er an einer Tivoli Workload Scheduler-Workstation angemeldet ist.

group=*gruppenname***[,...]**
Nur für UNIX-Benutzer. Für Benutzer der Job Scheduling Console darf dieses Argument nicht verwendet werden. Gibt die UNIX-Gruppe an, der der Benutzer angehört. Platzhalterzeichen sind erlaubt.

logon=*benutzername*|*tme-admin*|**@ [,...]**
Dabei gilt Folgendes:

benutzername
Gibt den Namen an, unter dem der Benutzer an einer Tivoli Workload Scheduler-Workstation angemeldet ist. Platzhalterzeichen sind erlaubt. Für das Attribut **cpu=** muss ein Workstationname oder **@** angegeben werden.

tme-admin
Gibt den Namen der Tivoli-Administratorgruppe an, zu der der Benutzer gehört. Wenn der Name Leerzeichen enthält, muss er in doppelte Anführungszeichen gesetzt werden. Platzhalterzeichen sind erlaubt. Für das Attribut **cpu=** muss **$framework** oder **@** angegeben werden.

@ Gibt an, dass der Benutzer unter einem beliebigen Namen angemeldet ist oder zu einer beliebigen Tivoli-Administratorgruppe gehört.

objektart
Gibt die Objektart an, auf die der Benutzer zugreifen darf. Es gibt folgende Objektarten:

calendar
Benutzerkalender

cpu Workstations und Domänen

file Tivoli Workload Scheduler-Datenbankdateien

job Terminierte Jobs
parameter
Benutzerparameter
prompt
Globale Eingabeaufforderungen
resource
Zeitplanungsressourcen
schedule
Jobnetze
userobj
Benutzerobjekte

Sie können mehrere Objektarten in eine Benutzerdefinition aufnehmen. Auf Objektarten, die in der Definition nicht vorhanden sind, ist kein Zugriff möglich.

objektattribute
Gibt mindestens ein Attribut an, mit dem eine Gruppe der angegebenen Art festgelegt wird. Werden keine Attribute angegeben, kann auf alle Objekte dieser Art zugegriffen werden. Die Objektattribute werden wie folgt angegeben:

objektattribut[{+ | ~}*objektattribut*[...]]

Mit einem Pluszeichen (+) werden die Attribute hinzugefügt, über die die Objekte verfügen müssen. Mit einem Tildezeichen (~) werden die Attribute hinzugefügt, über die die Objekte nicht verfügen dürfen. Es gibt folgende Objektattribute:

- Für die Objektart **calendar**:

  name=*kalendername***[,...]**
  Gibt einen oder mehrere Kalendernamen an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Kalender.

- Für die Objektart **cpu** (Workstation):

  cpu=*workstation***[,...]**
  Gibt mindestens einen Workstation- oder Domänennamen an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

- Für die Objektart **file**:

  name=*dateiname***[,...]**
  Gibt die Namen von Tivoli Workload Scheduler-Datenbankdateien an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Dateien. Es gibt folgende Dateinamen:
  calendars
  Enthält Kalender.
  cpudata
  Enthält Workstations und Domänen.
  jobs Enthält Jobs.

mastsked
Enthält Jobnetze.
parameters
Enthält Parameter.
prodsked
Enthält den Produktionszeitplan.
prompts
Enthält Eingabeaufforderungen.
resources
Enthält Ressourcen.
security
Die Sicherheitsdatei.
Symphony
Enthält den Produktionsplan.

- Für die Objektart **job**:

  cpu=*workstation*
  Gibt den Namen der Workstation an, auf der der Job ausgeführt wird. Platzhalterzeichen sind erlaubt. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

  jcl=*″pfad″ | ″befehl″*
  Gibt den Befehl oder den Pfadnamen für die ausführbare Datei des Jobs an. Der Befehl bzw. Pfad muss in Anführungszeichen (″) gesetzt werden. Platzhalterzeichen sind erlaubt. Wird nichts angegeben, bezieht sich die Definition auf alle Jobdateien und Befehle.

  logon=*benutzername***[,...]**
  Gibt die Benutzernamen an, unter denen die Jobs ausgeführt werden. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, gilt die Definition für alle Benutzernamen. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$jclowner**, **$owner** und **$user**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

  name=[*jobnetz***.]***job***[,...]**
  Gibt den Namen des Tivoli Workload Scheduler-Jobs an. Die Angabe des Jobnetzes, in dem sich der Job befindet, ist optional. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, gilt die Definition für alle Jobnamen.

- Für die Objektart **parameter**:

  cpu=*workstation*
  Gibt die Workstation an, auf der die Parameter definiert sind. Platzhalterzeichen sind erlaubt. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

name=*parameter***[,...]**
Gibt einen oder mehrere Parameter an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Parameter.

- Für die Objektart **prompt**:

  name=*eingabeaufforderung***[,...]**
  Gibt mindestens eine Eingabeaufforderung an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Eingabeaufforderungen.

- Für die Objektart **resource**:

  cpu=*workstation***[,...]**
  Gibt den Namen der Workstation an, auf der die Ressourcen definiert sind. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

  name=*ressource***[,...]**
  Gibt einen oder mehrere Ressourcennamen an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Ressourcen.

- Für die Objektart **schedule** (Jobnetz):

  cpu=*workstation*
  Gibt den Namen der Workstation an, auf der die Jobnetze ausgeführt werden. Platzhalterzeichen sind erlaubt. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

  name=*jobnetz***[,...]**
  Gibt mindestens einen Jobnetznamen an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, gilt die Definition für alle Jobnetze.

- Für die Objektart **userobj**:

  cpu=*workstation*
  Gibt die Workstation an, auf der der Benutzer definiert ist. Platzhalterzeichen sind erlaubt. Erfolgt keine Angabe, bezieht sich die Definition auf alle Workstations. Folgende Tivoli Workload Scheduler-Variablen können verwendet werden: **$master**, **$remotes**, **$slaves** und **$thiscpu**. Weitere Informationen finden Sie unter „Von Tivoli bereitgestellte Variablen” auf Seite 59.

  logon=*benutzername***[,...]**
  Gibt einen oder mehrere Benutzernamen an. Platzhalterzeichen sind erlaubt. Mehrere Namen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, bezieht sich die Definition auf alle Benutzer.

aktion Gibt die Aktionen an, die Benutzer ausführen können. Mehrere Aktionen müssen durch Kommas getrennt werden. Erfolgt keine Angabe, sind keine Aktionen erlaubt. Die Eingabe von **access=@** gibt Benutzern die Möglichkeit, alle Aktionen auszuführen.

- Für die Objektart **calendar**:

  add Neue Kalender der Datenbank hinzufügen und speichern

  delete Kalender aus der Datenbank löschen

  display
  Kalender in der Datenbank anzeigen

  modify
  Kalender in der Datenbank ändern

  use Kalender zur Terminierung von Jobnetzen verwenden

- Für die Objektart **cpu**, zu der Workstations und Domänen gehören:

  add Neue Workstations und Domänen der Datenbank hinzufügen und speichern

  console
  Tivoli Workload Scheduler-Konsole anzeigen und ändern

  delete Workstations und Domänen aus der Datenbank löschen

  display
  Workstations und Domänen in der Datenbank anzeigen

  fence Jobgrenzen für Workstations im Produktionsplan ändern

  limit Jobbegrenzungen für Workstations im Produktionsplan ändern

  link Workstationverbindungen herstellen

  modify
  Workstations und Domänen in der Datenbank ändern und ersetzen

  shutdown
  Tivoli Workload Scheduler herunterfahren. Diese Aktion kann nur in der Befehlszeile ausgeführt werden.

  start Tivoli Workload Scheduler-Verarbeitung starten

  stop Tivoli Workload Scheduler-Verarbeitung stoppen

  unlink
  Workstationverbindungen trennen

  Damit ein Benutzer die Domänenmanagerfunktion auf eine Workstation übertragen kann, muss er über die Zugriffsberechtigungen **start** und **stop** auf der Workstation verfügen.

- Für die Objektart **file**:

  build Tivoli Workload Scheduler-Datenbankdateien erstellen. Diese Aktion kann nur in der Befehlszeile ausgeführt werden.

  delete Ist noch nicht implementiert.

  display
  Zugriff auf die Sicherheitsdatei mit dem Befehl **dumpsec**. Außerdem

können Kalender, Parameter, Eingabeaufforderungen und Ressourcenstammdateien angezeigt werden. Diese Aktionen können nur in der Befehlszeile ausgeführt werden.

modify
Zugriff auf die Sicherheitsdatei mit dem Befehl **makesec**. Außerdem können Änderungen an Kalendern, Parametern, Eingabeaufforderungen und Ressourcenstammdateien vorgenommen werden. Diese Vorgänge können nur in der Befehlszeile ausgeführt werden.

Anmerkung: Mit der Objektart **file** wird die Sicherheit in der CLI verwaltet; diese Art ist nur für die CLI gültig.

- Für die Objektart **job**:

  add Neue Jobs der Datenbank hinzufügen und speichern

  adddep
  Abhängigkeiten für Jobs im Produktionsplan hinzufügen. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  altpri Jobprioritäten im Produktionsplan ändern. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  cancel Jobs im Produktionsplan abbrechen. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  confirm
  Die Beendigung von Jobs im Produktionsplan bestätigen. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  deldep
  Jobabhängigkeiten aus dem Produktionsplan löschen. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  delete Jobs aus der Datenbank löschen

  display
  Jobs in der Datenbank anzeigen

  kill Jobs im Produktionsplan mit dem Befehl KILL beenden

  modify
  Jobs in der Datenbank ändern und ersetzen

  release
  Jobabhängigkeiten im Produktionsplan freigeben. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

  reply Auf Jobeingabeaufforderungen im Produktionsplan antworten

  rerun Die Ausführung von Jobs im Produktionsplan wiederholen. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

submit
Jobs an den Produktionsplan übergeben. Diese Aktion ist nicht auf Workstations verfügbar, die mit Tivoli Workload Scheduler for z/OS für die Endpunkt-zu-Endpunkt-Zeitplanung verbunden sind.

use Jobs den Jobnetzen in der Datenbank hinzufügen

- Für die Objektart **parameter**:

  add Neue Parameter der Datenbank hinzufügen und speichern

  delete Parameter aus der Datenbank löschen

  display
  Parameter in der Datenbank anzeigen

  modify
  Parameter in der Datenbank ändern und ersetzen

- Für die Objektart **prompt**:

  add Neue Eingabeaufforderungen der Datenbank hinzufügen und speichern

  delete Eingabeaufforderungen aus der Datenbank löschen

  display
  Eingabeaufforderungen in der Datenbank anzeigen

  modify
  Eingabeaufforderungen in der Datenbank ändern und ersetzen

  use Eingabeaufforderungen Jobnetzen in der Datenbank hinzufügen und Eingabeaufforderungen Jobs und Jobnetzen im Produktionsplan hinzufügen

- Für die Objektart **resource**:

  add Neue Ressourcen der Datenbank hinzufügen und speichern

  delete Ressourcen aus der Datenbank löschen

  display
  Ressourcen in der Datenbank anzeigen

  modify
  Ressourcen in der Datenbank ändern und ersetzen

  use Ressourcen Jobnetzen in der Datenbank und Ressourcen Jobs und Jobnetzen im Produktionsplan hinzufügen

- Für die Objektart **schedule** (Jobnetz):

  add Neue Jobnetze der Datenbank hinzufügen und speichern

  adddep
  Abhängigkeiten Jobnetzen im Produktionsplan hinzufügen

  altpri Die Priorität von Jobnetzen im Produktionsplan ändern

  cancel Jobnetze im Produktionsplan abbrechen

  deldep
  Jobnetzabhängigkeiten im Produktionsplan löschen

  delete Jobnetze aus der Datenbank löschen

display
Jobnetze in der Datenbank anzeigen

limit Jobbegrenzungen von Jobnetzen im Produktionsplan ändern

modify
Jobnetze in der Datenbank ändern und ersetzen

release
Jobnetzabhängigkeiten im Produktionsplan freigeben

reply Auf Eingabeaufforderungen von Jobnetzen im Produktionsplan antworten

submit
Jobnetze an den Produktionsplan übergeben

- Für die Objektart **userobj**:

  add Neue Benutzer der Datenbank hinzufügen

  delete Benutzer aus der Datenbank löschen

  display
  Benutzer in der Datenbank anzeigen

  modify
  Benutzer in der Datenbank ändern und ersetzen

  altpass
  Benutzerkennwörter in der Datenbank ändern

Reihenfolge für Benutzerqualifikation

Wenn Benutzern die Berechtigung für den Zugriff auf Tivoli Workload Scheduler-Objekte erteilt wird, werden die tatsächlichen Attribute des Benutzers mit den Benutzerdefinitionen verglichen, und zwar in der Reihenfolge, in der die Definitionen in der Datei **Security** vorkommen. Die Berechtigungen des Benutzers werden durch die erste Definition festgelegt, die mit den Attributen des Benutzers übereinstimmt. Daher sollten die Benutzerdefinitionen so angeordnet werden, dass Definitionen mit den allgemeinsten Berechtigungen am Ende, Definitionen mit sehr spezifischen Berechtigungen am Anfang stehen. Beispiel:

```
#Falsch:
CPU=@+LOGON=TWSUser

#Richtig:
CPU=@+LOGON=TWSDomain\TWSUser
```

Weitere Informationen finden Sie unter „Beispiel der Datei ″Security″” auf Seite 60.

Reihenfolge für Objektqualifikation

In einer Benutzerdefinition können mit mehreren Anweisungen für eine einzige Objektart verschiedene Zugriffsberechtigungen für verschiedene Objektgruppen erteilt werden. Da die erste übereinstimmende Anweisung verwendet wird, ist die Reihenfolge der Objektanweisungen von Bedeutung. Am Anfang müssen die Anweisungen für sehr spezifische Berechtigungen, am Ende Anweisungen für allgemeinere Berechtigungen stehen. Beispiel:

```
#Falsch:
job name=@ access=display
job name=ar@ access=@

#Richtig:
job name=ar@ access=@
job name=@ access=display
```

Weitere Informationen finden Sie unter „Beispiel der Datei ″Security″” auf Seite 60.

Von Tivoli bereitgestellte Variablen

Die folgenden von Tivoli bereitgestellten Variablen können in Objektattributen verwendet werden:

$jclgroup
Der Gruppenname der ausführbaren Datei eines Jobs

$jclowner
Der Eigner der ausführbaren Datei eines Jobs

$master
Der Tivoli Workload Scheduler-Hauptdomänenmanager

$owner
Der Ersteller eines Jobnetzes und der zugehörigen Jobs

$remotes
Alle Standardagentenworkstations

$slaves
Alle Workstations fehlertoleranter Agenten

$thiscpu
Die Workstation, auf der der Benutzer den Tivoli Workload Scheduler-Befehl bzw. das Tivoli Workload Scheduler-Programm ausführt

$user Der Benutzer, der den Tivoli Workload Scheduler-Befehl bzw. das Tivoli Workload Scheduler-Programm ausführt

Die Variablen **$jclgroup** und **$jclowner** sind nur überprüfbar, wenn der Benutzer ein Tivoli Workload Scheduler-Programm auf der Workstation ausführt, auf der sich die ausführbare Datei des Jobs befindet. Wenn das Programm auf einer anderen Workstation ausgeführt wird, erhält der Benutzer keinen Zugriff.

Platzhalterzeichen

Die folgenden Platzhalterzeichen sind erlaubt, wenn sie in den Syntaxbeschreibungen angegeben sind:

? Ersetzt ein alphabetisches Zeichen.

% Ersetzt ein numerisches Zeichen.

@ Ersetzt null oder mehr alphanumerische Zeichen.

Der Superuser unter UNIX

Falls keine Sicherheitsdatei vorhanden ist, kann nur der Benutzer **root** auf Tivoli Workload Scheduler-Objekte zugreifen; der Benutzer **root** verfügt über unbeschränkten Zugriff auf alle Objekte und kann alle Tivoli Workload Scheduler-Programme und Befehle ausführen. Zur Steuerung von **root** erstellen Sie eine Sicherheitsdatei mit einer Benutzerdefinition für den Benutzer **root**. In der Sicherheitsdatei für ein Netzwerk können Sie zwischen den lokalen Benutzern **root** und dem Benutzer **root** des Hauptdomänenmanagers unterscheiden. Sie können z. B. lokale Benutzer dahingehend einschränken, dass sie nur Operationen ausführen können, die ihre eigenen Anmeldeworkstations betreffen, und dem Benutzer **root** des Masters die Ausführung von Operationen gestatten, die sich auf jede beliebige Workstation im Netzwerk auswirken. Weitere Informationen finden Sie unter “Beispiel der Datei **Security**”.

Beispiel der Datei ″Security″

Nachfolgend wird ein Beispiel der Datei **Security** aufgeführt. Eine Erklärung der Datei folgt auf dieses Beispiel.

```
###########################################################
#     Sample Security File
###########################################################
# (1)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON THE
#      MASTER DOMAIN MANAGER OR FRAMEWORK.
user mastersm  cpu=$master,$framework + logon=maestro,root,Root_london-region
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job                           access=@
schedule                      access=@
resource                      access=@
prompt                        access=@
file                          access=@
calendar                      access=@
cpu                           access=@
parameter  name=@ ~ name=r@   access=@
userobj    cpu=@ + logon=@    access=@
end
###########################################################
# (2)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON ANY
#      WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER.
user sm  logon=maestro,root
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu     access=@
schedule      cpu=$thiscpu     access=@
resource      cpu=$thiscpu     access=@
prompt                         access=@
file                           access=@
calendar                       access=@
cpu           cpu=$thiscpu     access=@
parameter     cpu=$thiscpu
              ~ name=r@        access=@
end
```

```
###########################################################
# (3)  APPLIES TO USERS LOGGED INTO THE SYS GROUP ON THE
#      MASTER DOMAIN MANAGER OR FRAMEWORK.
user masterop  cpu=$master,$fw + group=sys
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=@
              + logon=TWSDomain\TWSUser access=@
job           cpu=@
              + logon=root     access=adddep,altpri,cancel,
                                      confirm,deldep,release,
                                      reply,rerun,submit,use
job           cpu=@
              + logon=$user,$jclowner
              ~ logon=root     access=add,adddep,altpri,
                                      cancel,confirm,
                                      deldep,release,reply,
                                      rerun,submit,use
schedule      cpu=$thiscpu     access=@
schedule      cpu=@            access=adddep,altpri,cancel,
                                      deldep,limit,release,
                                      submit
resource                       access=add,display,
                                      resource,use
prompt                         access=add,display,reply,use
file                           access=build
calendar                       access=display,use
cpu           cpu=@            access=@
parameter  name=@ ~ name=r@    access=@
end
###########################################################
# (4)  APPLIES TO USERS LOGGED INTO THE SYS GROUP ON ANY
#      WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER
user op  group=sys
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu
              + logon=$user    access=@
job           cpu=$thiscpu
              + logon=root     access=adddep,altpri,cancel,
                                      confirm,deldep,release,
                                      reply,rerun,submit,use
job           cpu=$thiscpu
              ~ logon=root     access=adddep,altpri,cancel,
                                      confirm,deldep,release,
                                      reply,rerun,submit,use
schedule      cpu=$thiscpu     access=@
resource                       access=add,display,resource,use
prompt                         access=add,display,reply,use
file                           access=build
calendar                       access=use
cpu           cpu=$thiscpu     access=console,fence,limit,
                                      link,start,stop,unlink
parameter  name=@ ~ name=r@    access=@
end
```

```
############################################################
# (5)  APPLIES TO USERS LOGGED INTO THE MIS GROUP ON
#      ANY WORKSTATION OR FRAMEWORK.
user misusers  group=mis
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu
              + logon=$user    access=@
job           cpu=$thiscpu
              + logon=$jclowner
              ~ logon=root     access=submit,use
schedule      cpu=$thiscpu     access=add,submit,
                                      modify,display
parameter     name=r@          access=@
parameter     name=@           access=display
end
############################################################
# (6)  APPLIES TO ALL OTHER USERS LOGGED IN ON ANY
#      WORKSTATION.
user default  logon=@
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu
              + logon=$user    access=@
job           cpu=$thiscpu
              + logon=$jclowner
              ~ logon=root     access=submit,use
schedule      cpu=$thiscpu     access=add,submit,
                                      modify,display
parameter     name=u@          access=@
parameter     name=@ ~ name=r@ access=display
end
############################################################
```

Erklärung des Beispiels der Datei ″Security″

Beachten Sie bei der Reihenfolge der Definitionen Folgendes: die spezifischste Definition steht am Anfang und die allgemeinste am Ende. Aufgrund der Reihenfolge werden die Benutzer **maestro** und **root** zuerst abgeglichen, gefolgt von den Benutzern der Gruppe **sys** und schließlich denen der Gruppe **mis**. Alle anderen Benutzer werden mit der letzten Definition abgeglichen, die am wenigsten spezifisch ist.

 # (1) APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON THE MASTER DOMAIN MANAGER OR FRAMEWORK. user mastersm cpu=$master,$fw + logon=maestro,root,Root_london-region

Diese Benutzerdefinition gilt für den GUI- und CLI-Zugriff der Benutzer **maestro** und **root**, die an einem Hauptdomänenmanager angemeldet sind. Ebenso erteilt sie den Benutzern den GUI-Zugriff, die in der Tivoli-Administratorengruppe **Root_london-region** oder auf einem Management Framework-Computer aufgelistet sind. Sie erhalten den unbeschränkten Zugriff auf alle Objekte, mit Ausnahme der Parameter, deren Namen mit dem Buchstaben **r** beginnen. Der Zugriff auf die Parameter, deren Name mit **r** beginnt, wird nur den Benutzern der Gruppe **mis** erteilt.

 # (2) APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON ANY WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER. user sm logon=maestro,root

Diese Benutzerdefinition gilt für die Benutzer **maestro** und **root**, auf die die Definition 1 nicht zutrifft. Dies sind diejenigen, die an einer beliebigen Workstation, mit Ausnahme des Hauptdomänenmanagers oder eines Management Framework-Computers, angemeldet sind. Sie erhalten unbeschränkten Zugriff auf alle Objekte auf ihrer Anmeldeworkstation. Beachten Sie, dass Eingabeaufforderungen, Dateien und Kalender globaler Natur sind und nicht einer Workstation zugeordnet sind.

 # (3) APPLIES TO USERS LOGGED INTO THE SYS GROUP ON THE MASTER DOMAIN MANAGER OR FRAMEWORK. user masterop cpu=$master,$framework + group=sys

Diese Benutzerdefinition gilt für Benutzer, die an der Gruppe **sys** des Hauptdomänenmanagers oder eines Management Framework-Computers angemeldet sind. Sie erhalten eine eindeutige Gruppe von Zugriffsberechtigungen. Mehrere Objektanweisungen werden verwendet, um diesen Benutzern spezifische Arten des Zugriffs auf unterschiedliche Gruppen von Objekten zu geben. Es gibt z. B. drei Jobanweisungen:

- Die erste Jobanweisung gestattet den unbeschränkten Zugriff auf Jobs, die auf beliebigen Workstations (@) unter dem Namen des Benutzers (**$user**) ausgeführt werden.
- Die zweite Jobanweisung gestattet spezifische Arten des Zugriffs auf Jobs, die auf beliebigen Workstations und unter **root** ausgeführt werden.
- Die dritte Jobanweisung gestattet spezifische Arten des Zugriffs auf Jobs, die auf beliebigen Workstations ausgeführt werden. Die Jobs müssen unter dem Namen des Benutzers (**$user**) oder unter dem Namen des Jobdateieigners (**$jclowner**) ausgeführt werden. Jobs, die unter **root** ausgeführt werden, sind ausgeschlossen.

 # (4) APPLIES TO USERS LOGGED INTO THE SYS GROUP ON ANY WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER. user op group=sys

Diese Benutzerdefinition gilt für die Benutzer der Gruppe **sys**, auf die die Definition 3 nicht zutrifft. Dies sind diejenigen, die an einer beliebigen Workstation, mit Ausnahme des Hauptdomänenmanagers oder eines Management Framework-Computers, angemeldet sind. Sie erhalten eine Reihe von Zugriffsberechtigungen, die denen der Definition 3 gleichen. Es gibt folgende Ausnahme: Der Zugriff ist auf die Objekte auf der Anmeldeworkstation (**$thiscpu**) des Benutzers beschränkt.

 # (5) APPLIES TO USERS LOGGED INTO THE MIS GROUP ON ANY WORKSTATION OR FRAMEWORK. user misusers group=mis

Diese Benutzerdefinition gilt für Benutzer, die an der Gruppe **mis** beliebiger Workstations oder eines Management Framework-Computers angemeldet sind. Sie erhalten eine begrenzte Gruppe von Zugriffsmöglichkeiten. Ressourcen, Eingabeaufforderungen, Dateien, Kalender und Workstations werden nicht angegeben, wodurch der Zugriff auf diese Objekte verhindert wird. Diese Benutzer erhalten unbeschränkten Zugriff auf die Parameter, deren Namen mit dem Buchstaben **r** beginnen, andere Parameter können sie jedoch nur anzeigen.

 # (6) APPLIES TO ALL OTHER USERS LOGGED IN ON ANY WORKSTATION.

user default logon=@

Diese Benutzerdefinition erteilt Benutzern, die von den vorangegangenen Definitionen noch nicht abgedeckt wurden, eine Gruppe von Standardberechtigungen. Diese Benutzer erhalten unbeschränkten Zugriff auf die Parameter, deren Namen mit dem Buchstaben **u** beginnen, andere Parameter können sie jedoch nur anzeigen. Sie erhalten keinen Zugriff auf Parameter, deren Namen mit dem Buchstaben **r** beginnen.

Der Befehl ″dumpsec″

Dekompiliert die Datei **Security** und sendet die Ausgabe an **stdout**.

Der Benutzer muss die Zugriffsberechtigung **display** für die Datei **Security** haben.

Syntax

dumpsec -v | -u

dumpsec *sicherheitsdatei*

Beschreibung

Wenn keine Argumente angegeben sind, wird ein Speicherauszug der aktiven Sicherheitsdatei (**../unison/Security**) erstellt. Wenn Sie eine editierbare Kopie der Sicherheitsdatei erstellen wollen, leiten Sie die Ausgabe des Befehls in eine andere Datei, wie in den Beispielen gezeigt wird.

Argumente

-v Zeigt nur die Versionsinformationen zum Befehl an.

-u Zeigt nur Syntaxinformationen zum Befehl an.

sicherheitsdatei
Gibt den Namen der Sicherheitsdatei an, von der ein Speicherauszug erstellt werden soll.

Beispiele

Im folgenden Beispiel wird die Befehlsversion angezeigt:

```
dumpsec -v
```

Im folgenden Beispiel wird ein Speicherauszug der aktiven Sicherheitsdatei an **stdout** ausgegeben:

```
dumpsec
```

Im folgenden Beispiel wird ein Speicherauszug der aktiven Sicherheitsdatei an eine Datei namens **mysec** ausgegeben:

```
dumpsec > mysec
```

Im folgenden Beispiel wird ein Speicherauszug einer Sicherheitsdatei namens **sectemp** an **stdout** ausgegeben:

```
dumpsec sectemp
```

Der Befehl ″makesec″

Kompiliert die Benutzerdefinitionen und installiert die Sicherheitsdatei. Die Änderungen an der Sicherheitsdatei werden erst wirksam, nachdem Sie Tivoli Workload Scheduler gestoppt und erneut gestartet haben. Davon betroffen sind folgende Komponenten:

- conman
- composer
- Tivoli Workload Scheduler-Connector

Sie müssen die Programme nur beenden. Wenn sie das nächste Mal ausgeführt werden, werden die neuen Sicherheitsdefinitionen erkannt. Tivoli Workload Scheduler-Connector müssen Sie stoppen, indem Sie den Befehl **wmaeutil**

ausführen. Die Connector werden automatisch bei der Aktualisierung einer beliebigen Abfrage in der Job Scheduling Console neu gestartet.

Der Benutzer muss die Zugriffsberechtigung **modify** für die Sicherheitsdatei haben.

Anmerkung: Bei Windows NT müssen die Connector-Prozesse mit Hilfe des Befehls **wmaeutil** gestoppt werden, bevor der Befehl **makesec** ordnungsgemäß funktioniert.

Syntax

makesec -v | -u

makesec [**-verify**] *eingabedatei*

Beschreibung

Der Befehl **makesec** kompiliert die angegebene Datei und installiert sie als aktive Sicherheitsdatei (**../unison/Security**). Falls das Argument **-verify** angegeben ist, wird die Datei auf korrekte Syntax hin überprüft, sie wird allerdings nicht kompiliert und installiert.

Argumente

-v Zeigt nur die Versionsinformationen zum Befehl an.

-u Zeigt nur Syntaxinformationen zum Befehl an.

-verify
Überprüft nur die Syntax der Benutzerdefinitionen in der Eingabedatei. Die Datei wird nicht als die Sicherheitsdatei installiert. (Die Syntaxüberprüfung wird automatisch ausgeführt, wenn die Sicherheitsdatei installiert wird.)

eingabedatei
Gibt den Namen einer Datei oder einer Gruppe von Dateien an, in der/denen die Benutzerdefinitionen enthalten ist/sind. Ein Erweiterungsmuster für Dateinamen ist zugelassen.

Beispiele

Im folgenden Beispiel wird die Befehlsversion angezeigt:

```
makesec -v
```

Im folgenden Beispiel wird eine editierbare Kopie der aktiven Sicherheitsdatei in einer Datei namens **tempsec** erstellt; die Benutzerdefinitionen mit einem Texteditor geändert; dann wird die Datei **tempsec** kompiliert und die aktive Sicherheitsdatei ersetzt:

```
dumpsec > tempsec
edit tempsec
```

Hier führen Sie erforderliche Änderungen an der Datei **tempsec** durch. Wenn Sie mit den Änderungen an der Datei **tempsec** fertig sind, führen Sie den Befehl **makesec** aus, um die Sicherheitsdatei in Tivoli Workload Scheduler zu laden:

```
makesec tempsec
```

Im folgenden Beispiel werden Benutzerdefinitionen der Dateigruppe **userdef*** kompiliert und die aktive Sicherheitsdatei ersetzt:

```
makesec userdef*
```

5

Optionale Anpassungsoptionen

Nach der Installation des Produkts können Sie es anpassen, damit es Ihren Anforderungen entspricht. In diesem Kapitel werden die optionalen Anpassungsschritte für Ihre Tivoli Workload Scheduler-Installation beschrieben.

Globale Optionen

Die globalen Optionen sind auf dem Hauptdomänenmanager definiert. Sie gelten für alle Workstations im Tivoli Workload Scheduler-Netzwerk.

Globale Optionen setzen

Globale Optionen werden in der Datei **globalopts** mit Hilfe eines Texteditors eingegeben. Sie können jederzeit Änderungen vornehmen, diese werden jedoch erst nach einem Stopp und Neustart von Tivoli Workload Scheduler wirksam. Die Syntax wird in der nachfolgenden Tabelle beschrieben. Bei den Einträgen muss die Groß- und Kleinschreibung nicht beachtet werden.

Syntax der globalen Optionen	Standardwert
# *kommentar*	
automatically grant logon as batch = yes|no	**no**
batchman schedule = yes|no	**no**
bmmsgbase = *ganze-zahl*	**1000**
bmmsgdelta = *ganze-zahl*	**1000**
carry job states = ([*status*[,...]]**)**	Null
carryforward = yes|no|all	**yes**
company = *firmenname*	Null
database audit level = 0|1	**0**
expanded version = yes|no	Null
history = *tage*	**10**
ignore calendars = yes|no	**no**
master =*workstation*	Wird standardmäßig von **customize** unter UNIX und von **Setup** unter Windows NT gesetzt
plan audit level = 0|1	**0**
retain rerun job name = yes|no	**no**
start =*startzeit*	**0600**
timezone enable = yes|no	**no**

 # kommentar
Alles zwischen Nummernzeichen (#) und Zeilenende wird als Kommentar behandelt.

automatically grant logon as batch job
Dies gilt nur für Windows NT-Jobs. Ist diese Option auf **yes** gesetzt, wird den sich für Windows NT-Jobs anmeldenden Benutzern automatisch die Berechtigung **Anmelden als Stapelverarbeitungsauftrag** erteilt. Ist sie auf **no** gesetzt oder wird sie übergangen, muss die Berechtigung jedem Benutzer bzw. jeder Gruppe manuell erteilt werden. Beachten Sie, dass die Berechtigung für Benutzer, die Jobs auf einem Sicherungsdomänencontroller ausführen, nicht automatisch erteilt werden kann, Sie müssen diese Berechtigungen daher manuell erteilen.

bmmsgbase
Gibt die maximale Anzahl Eingabeaufforderungen an, die dem Operator nach der abnormalen Beendigung eines Jobs angezeigt werden kann. Der Standardwert ist **1000**.

bmmsgdelta
Gibt eine zusätzliche Anzahl Eingabeaufforderungen für den in **bmmsgbase** definierten Wert für den Fall an, dass ein Job nach einer abnormalen Beendigung wiederholt wird und die in **bmmsgbase** angegebene Grenze erreicht wurde. Der Standardwert ist **1000**.

batchman schedule
Dies ist eine Produktionsoption, die sich auf die Funktionsweise von Batchman auswirkt; Batchman ist der Produktionssteuerungsprozess von Tivoli Workload Scheduler. Die Einstellung legt die Priorität fest, die den Jobnetzen für nicht terminierte Jobs zugeordnet ist. Geben Sie **yes** ein, um diesen Jobnetzen eine Priorität von **10** zuzuordnen. Geben Sie **no** ein, um diesen Jobnetzen eine Priorität von **0** zuzuordnen.

carry job states
Dies ist eine Vorbereitungsoption, die sich auf die Funktionsweise des Befehls **stageman** auswirkt. Die Einstellung legt anhand des Status fest, welche Jobs in weitergeleitete Jobnetze übernommen werden sollen. Sie müssen die Angabe des Jobstatus in runde Klammern, doppelte Anführungszeichen oder einfache Anführungszeichen setzen. Die Kommas können durch Leerzeichen ersetzt werden. Die folgenden internen Jobstatusangaben sind gültig:

abend	**abenp**	**add**	**done**	**exec**	**fail**
hold	**intro**	**pend**	**ready**	**rjob**	**sched**
skel	**succ**	**succp**	**susp**	**wait**	**waitd**

Einige Beispiele für die Option:

```
carry job states=(abend,exec,hold,intro)
carry job states="abend exec hold intro"
carry job states='abend exec hold intro'
```

Eine leere Liste wird wie folgt angegeben:

```
carry job states=()
```

Weitere Informationen finden Sie unter „Die Weiterleitungsoptionen” auf Seite 73.

carryforward
Dies ist eine Vorbereitungsoption, die sich auf die Funktionsweise des Befehls **stageman** auswirkt. Die Einstellung legt fest, ob nicht abgeschlossene Jobnetze aus dem alten an den neuen Produktionsplan (Symphony) weitergeleitet werden. Geben Sie **yes** ein, damit nicht abgeschlossene Jobnetze nur weitergeleitet werden, wenn die

Option **Weiterleiten** in der Jobnetzdefinition aktiviert ist. Geben Sie **all** ein, damit alle nicht abgeschlossenen Jobnetze ungeachtet der Option **Weiterleiten** weitergeleitet werden. Geben Sie **no** ein, um die Weiterleitungsfunktion vollständig zu inaktivieren. Der Befehlszeilenoption **stageman -carryforward** werden dieselben Werte zugeordnet und sie dient derselben Funktion wie die globale Option **carryforward**. Falls diese verwendet wird, überschreibt sie die globale Option. Weitere Informationen finden Sie unter „Die Weiterleitungsoptionen" auf Seite 73.

company

Dies ist der Name Ihrer Firma, er kann bis zu 40 Zeichen umfassen. Wenn der Name Leerzeichen enthält, setzen Sie den gesamten Namen in Anführungszeichen (″).

database audit level

Das Aktivieren bzw. Inaktivieren der Datenbankprüfung kann ausgewählt werden. Gültige Werte sind **0** für das Inaktivieren der Datenbankprüfung und **1** für das Aktivieren der Datenbankprüfung. Die Prüfinformationen werden in einer unstrukturierten Datei im Verzeichnis *TWShome*/**audit/database** protokolliert. Für jede Tivoli Workload Scheduler-Workstation wird ein eigenes Protokoll verwaltet. Für die Datenbank werden nur Aktionen und nicht das Delta der Aktion in der Prüfdatei protokolliert. Weitere Informationen zu dieser Funktion finden Sie unter ″Anhang A, „Zeitzonen und Prüfung"″.

expanded version

Diese Option wird während der Installation von **customize** unter UNIX und von **Setup** unter Windows NT gesetzt. Wenn die Option auf **yes** gesetzt ist, werden erweiterte Datenbanken verwendet. Wenn die Option auf **no** gesetzt ist, werden erweiterte Datenbanken nicht verwendet. Erweiterte Datenbanken ermöglichen die Verwendung langer Objektnamen. Erweiterte Jobnamen können z. B. bis zu sechzehn Zeichen enthalten. Die Option wird auch auf **yes** gesetzt, wenn Sie mit dem Dienstprogramm **dbexpand** nicht erweiterte Datenbanken in erweiterte Datenbanken umwandeln. Wenn die Option nicht vorhanden ist, wie dies in Versionen vor Maestro Version 6.0 der Fall ist, wird sie als **no** interpretiert.

history

Es kann eingegeben werden, wie viele Tage die statistischen Daten zu Jobs gespeichert werden sollen. Die Statistiken werden auf FIFO-Basis gelöscht. Diese Option wirkt sich nicht auf Standardlistdateien mit Jobs aus. Diese müssen mit dem Befehl **rmstdlist** entfernt werden. Informationen zum Befehl **rmstdlist** finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

ignore calendars

Dies ist eine Vorbereitungsoption, die sich auf die Funktionsweise des Befehls **compiler** auswirkt. Die Einstellung legt fest, ob Benutzerkalender in die neue Produktionssteuerungsdatei kopiert werden. Geben Sie **yes** ein, um das Kopieren von Benutzerkalendern in den neuen Produktionsplan (die Datei **Symphony**) zu verhindern. Dadurch wird Speicherplatz in der Datei freigehalten, aber die Verwendung von Kalendernamen in Datumsausdrücken zugelassen. Geben Sie **no** ein, damit Benutzerkalender in den neuen Produktionsplan kopiert werden. Weitere Informationen finden Sie in der Erklärung zum Befehl **compiler** im Handbuch *Tivoli Workload Scheduler Referenz*.

master

Der Name des Hauptdomänenmanagers. Dieser wird festgelegt, wenn Sie Tivoli Workload Scheduler mit **customize** (UNIX) oder **Setup** (Windows NT) installieren.

plan audit level
Das Aktivieren bzw. Inaktivieren der Planprüfung kann ausgewählt werden. Gültige Werte sind **0** für das Inaktivieren der Planprüfung und **1** für das Aktivieren der Planprüfung. Die Prüfinformationen werden in einer unstrukturierten Datei im Verzeichnis *TWShome*/**audit/plan** protokolliert. Für jede Tivoli Workload Scheduler-Workstation wird ein eigenes Protokoll verwaltet. Für den Plan werden nur Aktionen und nicht das erfolgreiche Ausführen bzw. das Fehlschlagen der Aktion in der Prüfdatei protokolliert. Weitere Informationen zu dieser Funktion finden Sie unter ″Anhang A, „Zeitzonen und Prüfung”″.

retain rerun job name
Dies ist eine Produktionsoption, die sich auf die Funktionsweise von Batchman auswirkt; Batchman ist der Produktionssteuerungsprozess von Tivoli Workload Scheduler. Die Einstellung legt fest, ob Jobs, die mit dem Conman-Befehl **rerun** wiederholt werden, ihren ursprünglichen Jobnamen behalten. Geben Sie **yes** ein, damit die wiederholten Jobs ihren ursprünglichen Jobnamen behalten. Geben Sie **no** ein, damit den wiederholten Jobs der unter **rerun from** angegebene Name zugeordnet wird.

start Die Startzeit des Tivoli Workload Scheduler-Verarbeitungstages kann im 24-Stunden-Format eingegeben werden: *hhmm* (**0000**-**2359**). Die Standardstartzeit ist 6:00 Uhr und die Standardstartzeit des Jobnetzes **final** ist 5:59 Uhr. Wenn Sie diese Option ändern, müssen Sie auch die Startzeit des Jobnetzes **final** ändern, die in der Regel auf eine Minute vor der Startzeit gesetzt wird.

timezone enable
Das Aktivieren bzw. Inaktivieren der Zeitzonenoption kann ausgewählt werden. Gültige Werte sind **yes** für das Aktivieren von Zeitzonen im Netzwerk und **no** für das Inaktivieren von Zeitzonen im Netzwerk. Standardmäßig ist die Zeitzonenoption bei der Installation von Tivoli Workload Scheduler deaktiviert. Wenn der Eintrag **timezone enable** in der Datei **globalopts** fehlt, ist die Zeitzonenoption inaktiviert. Weitere Informationen zu dieser Funktion finden Sie unter ″Anhang A, „Zeitzonen und Prüfung”″.

Beispiel der Datei für globale Optionen

Eine Schablone der Datei für globale Optionen mit den Tivoli Workload Scheduler-Standardeinstellungen finden Sie unter *TWShome*/**config/globalopts**.

Eine Arbeitskopie der Datei für globale Optionen wird bei der Installation als *TWShome*/**mozart/globalopts** installiert.

Sie können die Arbeitskopie entsprechend Ihren Bedürfnissen anpassen. Nachfolgend wird ein Beispiel der Datei für globale Optionen aufgeführt.

```
# Globalopts file on the master domain manager defines
# attributes of the Tivoli Workload Scheduler network.
#--------------------------------------------------------
company="Tivoli Systems"
master=main
start=0600
history=10
carryforward=yes
ignore calendars=no
batchman schedule=no
retain rerun job name=no
#
#--------------------------------------------------------
# End of globalopts.
```

Die Weiterleitungsoptionen

Jobnetze werden durch den Befehl **stageman** während der Tagesabschlussverarbeitung weitergeleitet. Der Weiterleitungsprozess wird durch Folgendes beeinflusst:

- Das Schlüsselwort **carryforward** in Ihren Jobnetzen. Weitere Informationen finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.
- Die globale Option **carryforward**. Siehe Seite 70.
- Die Befehlszeilenoption **stageman -carryforward**. Weitere Informationen finden Sie in der Erklärung zum Befehl **stageman** im Handbuch *Tivoli Workload Scheduler Referenz*.
- Die globale Option **carry job states**. Siehe Seite 70.

In der folgenden Tabelle wird gezeigt, was die verschiedenen Kombinationen der Weiterleitungsoptionen bewirken.

Globale Optionen	Weiterleitungsoperation
carryforward=no	Es werden keine Jobnetze weitergeleitet.
carryforward=yes carry job states=(*status***)**	Jobnetze werden nur weitergeleitet, wenn sich ihre Jobs in den angegebenen Status befinden und die Weiterleitungsoption aktiviert ist. Es werden nur die Jobs in den angegebenen Status mit den Jobnetzen weitergeleitet.
carryforward=yes carry job states=()	Jobnetze werden nur weitergeleitet, wenn sie nicht abgeschlossen sind und die Weiterleitungsoption aktiviert ist. Es werden alle Jobs mit den Jobnetzen weitergeleitet.
carryforward=all carry job states=(*status***)**	Jobnetze werden nur weitergeleitet, wenn sich ihre Jobs in den angegebenen Status befinden. Es werden nur Jobs in den angegebenen Status mit den Jobnetzen weitergeleitet.
carryforward=all carry job states=()	Jobnetze werden nur weitergeleitet, wenn sie nicht abgeschlossen sind. Es werden alle Jobs mit den Jobnetzen weitergeleitet.

Nachfolgend sind Informationen zum Verhalten der Weiterleitungsoptionen aufgeführt:

- Jedes Jobnetz, das nicht im Status SUCC ist, wird als nicht abgeschlossen betrachtet und weitergeleitet.
- Die Befehlszeilenoption **stageman -carryforward**, falls verwendet, setzt immer die globale Option **carryforward** außer Kraft. Ist keine von beiden Optionen angegeben, ist die Standardeinstellung **carryforward=yes**.
- Der Standardeintrag für die globale Option **carry job states** ist null. Das heißt, wenn die Liste leer ist oder die Option nicht vorhanden, funktioniert das Weiterleiten wie für **carry job states=()** beschrieben.
- Jobs und Jobnetze, die abgebrochen wurden, werden nie weitergeleitet.
- Jobs und Jobnetze mit abgelaufenen **until**-Zeiten werden nie weitergeleitet.
- Die Entscheidung für das Weiterleiten eines sich wiederholenden Jobs (definiert durch die Option **Every**) hängt von dem Status seiner letzten Ausführung ab.
- Ist ein Job aktiv, wenn die Ausführung des Jobs **Jnextday** beginnt, und wurde er nicht für die Weiterleitung angegeben, wird der Job weiterhin ausgeführt und in das Jobnetz **userjobs** für den neuen Produktionstag gestellt. Beachten Sie, dass Abhängigkeiten von solchen Jobs nicht weitergeleitet und die von diesem Job gehaltenen Ressourcen freigegeben werden.

Globale Optionen für MPE-Agenten setzen

In einem Tivoli Workload Scheduler-Netzwerk mit einem als Domänenmanager konfigurierten UNIX- oder NT-System und mit als fehlertolerante Agenten (FTAs) konfigurierten HP3000-Systemen (MPE) können Sie eine Gruppe globaler Optionen auf dem UNIX/NT-Hauptdomänenmanager angeben, um die Funktionsweise von Tivoli Workload Scheduler auf den MPE-FTAs zu steuern.

Diese Optionen nehmen den Platz der Parameter ein, die andernfalls auf den MPE-Systemen mit Hilfe der CTP1-Transaktion des Programms Arranger gesetzt werden würden. Wenn Sie diese Optionen integrieren wollen, fügen Sie sie Ihrer Datei **globalopts** mit der in der folgenden Tabelle beschriebenen Syntax hinzu. Bei den Einträgen muss die Groß- und Kleinschreibung nicht beachtet werden.

Syntax der globalen Optionen	Standardwert
rules mode = yes|no	**no**
batchman schedule = yes|no	**no**
all userjobs in userjobs schedule = yes|no	**no**
set mpe job pri to zero = yes|no	**no**

rules mode
Ersetzt CTP1-Parameter 4, das ist **Complete Control Mode**. Wenn Sie diese Option auf **yes** setzen, müssen Sie auch die Option **batchman schedule** auf **yes** setzen. Der normale Status von Batchman kann **Aktiv** oder **Inaktiv** sein. Dies wird im Fenster für die Scheduler-Merkmale der Job Scheduling Console angezeigt oder wenn Sie den Befehl **conman status** in der CLI ausführen. Durch das Aktivieren dieser Option wird der aktive Status von Batchman so geändert, dass **Regeln** angezeigt wird.

batchman schedule
Ersetzt CTP1-Parameter 7, das ist **Assign priority 10 to Batchman-created job streams**. Beachten Sie, dass diese Option für UNIX und Windows NT gleichermaßen gültig ist. Siehe „Globale Optionen” auf Seite 69.

all userjobs in userjobs schedule
Ersetzt CTP1-Parameter 8, das ist **Place all user jobs in USERJOBS job stream**. Setzen Sie diese Option auf **no**, falls die Option **rules mode** auf **yes** gesetzt ist.

set mpe job pri to zero
Ersetzt CTP1-Parameter 9, das ist **Force MPE priority to 0 for all userjobs**. Setzen Sie diese Option auf **no**, falls die Option **all userjobs in userjobs schedule** auf **yes** gesetzt ist.

Lokale Optionen

Lokale Optionen werden auf jeder Workstation definiert und gelten nur für die jeweilige Workstation.

Lokale Optionen setzen

Sie können die lokalen Optionen in eine Datei namens **localopts** mit Hilfe eines Texteditors eingeben. Es können jederzeit Änderungen vorgenommen werden, diese werden jedoch erst nach einem Stopp und Neustart von Tivoli Workload Scheduler wirksam. Die Syntax wird in der nachfolgenden Tabelle beschrieben. Bei den Einträgen muss die Groß- und Kleinschreibung nicht beachtet werden.

Syntax der lokalen Optionen	Standardwert
# kommentar	
bm check file = *sekunden*	**120**
bm check status = *sekunden*	**300**
bm check until = *sekunden*	**300**
bm look = *sekunden*	**30**
bm read = *sekunden*	**15**
bm stats = **on**|**off**	**off**
bm verbose = **on**|**off**	**off**
composer prompt= *schlüssel*	**- (Bindestrich)**
conman cli prompt= *schlüssel*	**% (Prozent)**
date format= *ganze-zahl*	**1**
jm job table size = *einträge*	**160**
jm look = *sekunden*	**300**
jm nice = *wert*	**0**
jm no root = **yes**|**no**	**no**
jm read = *sekunden*	**10**
merge stdlists = **yes**|**no**	**yes**
mm cache mailbox= *yes*|*no*	no
mm cache size= *byte*	32
mm read = *sekunden*	**15**
mm response = *sekunden*	**600**
mm retry link = *sekunden*	**600**
mm sound off = **yes**|**no**	**no**
mm unlink = *sekunden*	**960**
mozart directory = *gemeinsames-mozart-verzeichnis*	Kein
nm ipvalidate = **none**|**full**	**none**
nm mortal = **yes**|**no**	**no**
nm port =*anschlussnummer*	**31111**
nm read = *sekunden*	**60**
nm retry = *sekunden*	**800**
parameters directory =*gemeinsames-parms-verzeichnis*	Kein
stdlist width = *spalten*	**80**
sync level= ***low***|***medium***|***high***	high
switch sym prompt=*schlüssel*	**% (Prozent)**
syslog local = *funktion*	**-1**
thiscpu = *workstation*	**thiscpu**
unison network directory = *gemeinsames-unison-verzeichnis*	Kein
wr read = *sekunden*	**600**
wr enable compression= *yes*|*no*	no
wr unlink = *sekunden*	**600**

 # kommentar
Alles zwischen Nummernzeichen (#) und Zeilenende wird als Kommentar behandelt.

bm check file
Gibt die Mindestwartezeit in Sekunden an, nach der Batchman überprüft, ob eine Datei vorhanden ist, die als Abhängigkeit verwendet wird.

bm check status
Gibt die Wartezeit in Sekunden zwischen der Überprüfung des Status einer INET-Abhängigkeit durch Batchman an.

bm check until
Gibt die maximale Wartezeit in Sekunden an, nach der Batchman das Verstreichen einer **Until**-Zeit für einen Job oder ein Jobnetz meldet. Die Angabe eines Werts, der unterhalb des Standardwerts (300) liegt, kann zu einer Überlastung des Systems führen. Wenn der Wert kleiner als der Wert der lokalen Option **bm read** ist, wird an seiner Stelle der Wert der Option **bm read** verwendet.

bm look
Gibt die Mindestwartezeit in Sekunden an, nach der Batchman die Produktionssteuerungsdatei durchsucht und aktualisiert.

bm read
Gibt die maximale Zeit in Sekunden an, die Batchman auf den Eingang einer Nachricht in der Nachrichtendatei wartet.

bm stats
Die Angabe des Werts **on** bewirkt, dass Batchman seine statistischen Daten zu Systemstart und Systemabschluss an seine Standardlistdatei sendet. Die Angabe des Werts **off** verhindert, dass Batchman seine statistischen Daten an seine Standardlistdatei sendet.

bm verbose
Die Angabe des Werts **on** bewirkt, dass Batchman alle Jobstatusnachrichten an seine Standardlistdatei sendet. Die Angabe des Werts **off** verhindert, dass die erweiterte Gruppe der Jobstatusnachrichten an die Standardlistdatei gesendet wird.

composer prompt
Gibt eine Eingabeaufforderung für die Composer-Befehlszeile an. Die Eingabeaufforderung kann aus bis zu 10 Zeichen bestehen. Der Standardwert ist ein Bindestrich (-).

conman prompt
Gibt eine Eingabeaufforderung für die Conman-Befehlszeile an. Die Eingabeaufforderung kann aus bis zu 8 Zeichen bestehen. Der Standardwert ist ein Prozentzeichen (%).

date format
Gibt den Wert an, der dem gewünschten Datumsformat entspricht. Folgende Werte sind möglich:

- 0 entspricht *jj/mm/tt*
- 1 entspricht *mm/tt/jj*
- 2 entspricht *tt/jj/mm*
- 3 gibt die Verwendung der Variablen für die Unterstützung in der Landessprache an.

Der Standardwert ist 1.

jm job table size
Gibt die Größe der von Jobman verwendeten Jobtabelle anhand der Anzahl der Einträge an.

jm look
Gibt die Mindestwartezeit in Sekunden an, nach der Jobman nach beendeten Jobs sucht und allgemeine Jobmanagementtasks ausführt.

jm nice
Nur für UNIX: Gibt an, dass der Wert **nice** auf die mit Jobman gestarteten Jobs angewendet werden soll.

jm no root
Nur für UNIX: Die Angabe des Werts **yes** verhindert, dass Jobman **root**-Jobs startet. Die Angabe des Werts **no** ermöglicht Jobman das Starten von **root**-Jobs.

jm read
Gibt die Zeit in Sekunden an, die Jobman höchstens auf den Eingang einer Nachricht in der Nachrichtendatei wartet.

mm cache mailbox
Mit dieser Option wird Mailman für die Verwendung eines Lesecache für ankommende Nachrichten aktiviert. In diesem Fall werden nur die Nachrichten zwischengespeichert, die für die Netzwerkdatenkonsistenz nicht essenziell sind. Der Standardwert ist **no**.

mm cache size
Diese Option sollte bei Verwendung der Option **mm cache mailbox** ebenfalls angegeben werden. Der Standardwert ist **32** Byte. Verwenden Sie den Standardwert für kleine und mittelgroße Netzwerke. Für größere Netzwerke sollten Sie größere Werte verwenden. Vermeiden Sie es, einen großen Wert für kleine Netzwerke anzugeben. Der Maximalwert lautet **512** (höhere Werte werden ignoriert).

merge stdlists
Durch die Angabe des Werts **yes** senden alle Tivoli Workload Scheduler-Steuerungsprozesse, mit Ausnahme von Netman, ihre Konsolnachrichten an eine einzige Standardlistdatei. Die Datei heißt **maestro**. Durch die Angabe des Werts **no** senden die Prozesse ihre Nachrichten an verschiedene Standardlistdateien.

mm read
Gibt das Zeitintervall in Sekunden an, in dem Mailman seine Mailbox auf Nachrichten überprüft. Der Standardwert ist **15** Sekunden. Die Angabe eines niedrigeren Werts bewirkt, dass Tivoli Workload Scheduler schneller ausgeführt wird, aber zugleich mehr Verarbeitungszeit in Anspruch nimmt.

mm response
Gibt die maximale Wartezeit in Sekunden an, die Mailman auf eine Antwort wartet, bevor gemeldet wird, dass eine Workstation nicht reagiert. Die Antwortzeit sollte nicht weniger als **90** Sekunden betragen.

mm retry link
Gibt die maximale Wartezeit in Sekunden an, nach der Mailman versucht, die Verbindung zu einer Workstation herzustellen, die zuvor aufgehoben wurde, weil die Workstation nicht mehr reagierte.

mm sound off
Gibt an, wie Mailman auf den Conman-Befehl **tellop ?** reagieren soll. Bei Angabe des Werts **yes** zeigt Mailman Informationen zu jeder von ihm ausgeführten Task an. Bei Angabe des Werts **no** sendet Mailman nur seinen eigenen Status.

mm unlink
Gibt die maximale Wartezeit in Sekunden an, nach der Mailman die Verbindung zu einer Workstation aufhebt, die nicht reagiert. Die Wartezeit sollte nicht kürzer als die Antwortzeit sein, die für die lokale Option **nm response** angegeben wurde.

nm ipvalidate
Durch die Angabe des Werts **full** wird die IP-Adressenüberprüfung aktiviert. Wenn die IP-Überprüfung fehlschlägt, ist die Verbindung nicht zulässig. Die Angabe des Werts **none** lässt Verbindungen zu, auch wenn die IP-Überprüfung fehlschlägt.

nm mortal
Durch die Angabe des Werts **yes** wird Netman nach dem Stoppen aller untergeordneten Prozesse beendet. Durch die Angabe des Werts **no** wird Netman selbst nach dem Stoppen aller untergeordneten Prozesse weiter ausgeführt.

nm port
Gibt die TCP-Anschlussnummer an, auf die Netman auf dem lokalen Computer reagiert. Diese muss mit dem TCP-Anschluss in der Workstationdefinition des Computers übereinstimmen. Dabei muss es sich um einen 16-Bit-Wert ohne Vorzeichen im Bereich von 1 bis 65535 handeln. (Beachten Sie, dass die Werte zwischen 0 und 1023 für Standarddienste, wie z. B. FTP, TELNET, HTTP usw., reserviert sind.)

nm read
Gibt die maximale Zeit in Sekunden an, die Netman auf eine Verbindungsanforderung wartet, bevor es seine Nachrichtenwarteschlange auf die Befehle **stop** und **start** überprüft.

nm retry
Gibt die maximale Wartezeit in Sekunden an, nach der Netman eine fehlgeschlagene Verbindung erneut herzustellen versucht.

stdlist width
Gibt die maximale Breite der Anzeige von Tivoli Workload Scheduler-Konsolnachrichten an. Sie können eine Spaltennummer im Bereich von **1** bis **255** angeben und die Zeilen werden an oder vor der angegebenen Spalte umbrochen, je nachdem, wo sich eingebettete Vorschubsteuerzeichen befinden. Geben Sie eine negative Zahl oder eine Null ein, damit die Zeilenlänge ignoriert wird. Unter UNIX sollten Sie die Zeilenlänge ignorieren, wenn Sie die Systemprotokollierung mit der Option **syslog local** aktivieren.

syslog local
Aktiviert bzw. inaktiviert die Tivoli Workload Scheduler-Systemprotokollierung ausschließlich für UNIX-Computer. Geben Sie den Wert **-1** an, um die Systemprotokollierung für Tivoli Workload Scheduler auszuschalten. Geben Sie eine Zahl zwischen **0** und **7** an, damit die Systemprotokollierung eingeschaltet wird und Tivoli Workload Scheduler die entsprechende lokale Funktion (LOCAL0 bis LOCAL7) für seine Nachrichten verwendet. Geben Sie eine beliebige andere Zahl ein, damit die Systemprotokollierung eingeschaltet wird und Tivoli Workload Scheduler die USER-Funktion für seine Nachrichten verwendet. Weitere Informationen finden Sie unter „Tivoli Workload Scheduler-Konsolnachrichten und -Eingabeaufforderungen” auf Seite 81.

sync level
Gibt die Geschwindigkeit der Schreibzugriffe auf der Platte an. Diese Option wirkt sich auf alle Mailbox-Agenten aus und gilt nur für UNIX-Workstations. Folgende Werte sind möglich:

low Überlässt dem Betriebssystem die Handhabung.

medium
Führt Aktualisierungen nach Beendigung einer Transaktion aus.

high Führt bei jeder Dateneingabe Aktualisierungen aus.

Der Standardwert ist **high**.

switch sym prompt
Gibt eine Eingabeaufforderung für die Conman-Befehlszeile an, nachdem eine andere Datei **Symphony** mit dem Befehl **setsym** ausgewählt wurde. Die Eingabeaufforderung kann aus bis zu 8 Zeichen bestehen. Der Standardwert ist ein Prozentzeichen (%).

thiscpu
Gibt den Tivoli Workload Scheduler-Namen dieser Workstation an.

wr enable compression
Diese Option wird auf fehlertoleranten Agenten verwendet. Gibt an, ob der fehlertolerante Agent die Datei **Symphony** in komprimierter Form vom Hauptdomänenmanager empfangen kann. Der Standardwert ist **no**.

wr read
Gibt die Zeit in Sekunden an, die der Writer-Prozess auf eine ankommende Nachricht wartet, bevor er überprüft, ob eine Beendigungsanforderung von Netman vorliegt.

wr unlink
Gibt die Wartezeit in Sekunden an, nach der der Writer-Prozess beendet wird, weil keine Nachrichten empfangen werden. Die Untergrenze liegt bei **120** Sekunden; der Standardwert ist **600**.

Beispiel der Datei für lokale Optionen

Eine Schablonendatei mit den Tivoli Workload Scheduler-Standardeinstellungen finden Sie unter *TWShome*/**config/localopts**.

Eine Arbeitskopie der Datei für lokale Optionen wird bei der Installation in *TWShome*/**localopts** installiert, es sei denn, Sie haben eine andere als die Standardposition für Netman angegeben. In dem Fall sind zwei Kopien der Datei **localopts** vorhanden: eine in *TWShome* und die andere in *Netmanhome*. Alle Netman betreffenden Optionen müssen in der Datei **localopts** im Verzeichnis *Netmanhome* aktualisiert werden.

Sie können die Arbeitskopie entsprechend Ihren Bedürfnissen anpassen. Nachfolgend wird ein Beispiel der Datei für lokale Optionen aufgeführt.

```
#
# Tivoli Workload Scheduler localopts file defines attributes of this Workstation.
#
#--------------------------------------------------------------------------
# Attributes of this Workstation:
#
thiscpu=<THISCPU>
merge stdlists=yes
stdlist width=80
syslog local=-1
```

```
#
#----------------------------------------------------------------------
# Attributes of this Workstation for Tivoli Workload Scheduler batchman process:
#
bm check file=120
bm check status=300
bm look=15
bm read=10
bm stats=off
bm verbose=off
bm check until=300
#
#----------------------------------------------------------------------
# Attributes of this Workstation for Tivoli Workload Scheduler jobman process:
#
jm job table size=1024
jm look=300
jm nice=0
jm no root=no
jm read=10
#
#----------------------------------------------------------------------
# Attributes of this Workstation for Tivoli Workload Scheduler mailman process:
#
mm response=600
mm retrylink                      =600
mm sound off                           =no
mm unlink=960
mm cache mailbox                       =no
mm cache size                          =32
#
#----------------------------------------------------------------------
# Attributes of this Workstation for Tivoli Workload Scheduler netman process:
#
nm mortal=no
nm port=<TCP_PORT>
nm read=10
nm retry=800
#
#----------------------------------------------------------------------
# Attributes of this Workstation for Tivoli Workload Scheduler writer process:
#
wr read                                =600
wr unlink                              =120
wr enable compression                  =no
#
#----------------------------------------------------------------------
# Optional attributes of this Workstation for remote database files
#
# mozart directory = <HOME>/mozart
# parameters directory = <HOME>
# unison network directory =  <HOME>/../unison/network
#
#----------------------------------------------------------------------
# Custom format attributes
#
date format =               1# The possible values are 0-ymd, 1-mdy, 2-dmy, 3-NLS.
composer prompt =           -
conman prompt =             %
switch sym prompt =        <n>%
#
#----------------------------------------------------------------------
# Attributes for customization of I/O on mailbox files
#
sync level =                high
#
#----------------------------------------------------------------------
```

Lokale Optionen für Netman setzen

Wenn das Netman-Basisverzeichnis ein anderes Verzeichnis als das Basisverzeichnis von Tivoli Workload Scheduler ist, werden die folgenden lokalen Optionen in eine separate Datei **localopts** im Verzeichnis von Netman versetzt:

- nm ipvalidate
- nm mortal
- nm port
- nm read
- nm retry
- merge stdlists
- stdlist width
- syslog local

Weitere Informationen zum Netman-Verzeichnis finden Sie unter „Basisverzeichnis von Netman" auf Seite 17.

Optionen für dezentralisierte Verwaltung unter Windows NT setzen

Wenn Sie Tivoli Workload Scheduler mit Hilfe der Prozedur installiert haben, die die dezentralisierte Verwaltung von Zeitplanungsobjekten zulässt, können Sie die gemeinsam benutzten Verzeichnisse auf dem Hauptdomänenmanager mit den folgenden Optionen definieren.

mozart directory
: Gibt den Namen des gemeinsam benutzten Verzeichnisses **mozart** auf dem Hauptdomänenmanager an.

unison network directory
: Gibt den Namen des gemeinsam benutzten Verzeichnisses **unison** auf dem Hauptdomänenmanager an.

parameters directory
: Gibt den Namen des gemeinsam benutzten Verzeichnisses *TWShome* auf dem Hauptdomänenmanager an.

Wird eine Option nicht gesetzt bzw. ist diese nicht vorhanden, versuchen die Tivoli Workload Scheduler-Programme, die Datenbankdateien auf dem lokalen Computer zu öffnen.

Tivoli Workload Scheduler-Konsolnachrichten und -Eingabeaufforderungen

Die Tivoli Workload Scheduler-Steuerungsprozesse (Netman, Mailman, Batchman, Jobman und Writer) schreiben ihre Statusnachrichten, auch Konsolnachrichten genannt, in Standardlistdateien. Diese Nachrichten schließen auch die Eingabeaufforderungen mit ein, die als Job- und Jobnetzabhängigkeiten verwendet werden. Unter UNIX können die Nachrichten auch an den Dämon **syslog** (**syslogd**) und an ein Terminal, auf dem der Tivoli Workload Scheduler Console Manager ausgeführt wird, geleitet werden. In den folgenden Abschnitten werden diese Funktionen beschrieben.

"syslog local" unter UNIX setzen

Wenn Sie **syslog local** in der Datei für die lokalen Optionen auf einen positiven Wert setzen, senden die Steuerungsprozesse von Tivoli Workload Scheduler ihre Konsol- und Eingabeaufforderungsnachrichten an den Dämon **syslog**. Durch das Setzen des Werts auf **-1** wird

diese Funktion inaktiviert. Wenn Sie die Option auf einen positiven Wert setzen, um die Systemprotokollierung zu aktivieren, müssen Sie auch die lokale Option **stdlist width** auf **0** oder einen negativen Wert setzen.

Die Konsolnachrichten von Tivoli Workload Scheduler entsprechen den folgenden **syslog**-Ebenen:

LOG_ERR
Fehlernachrichten, wie z. B. die abnormale Beendigung von Steuerungsprozessen und Dateisystemfehler

LOG_WARNING
Warnungen, wie z. B. Verbindungsfehler und blockierte Jobnetze

LOG_NOTICE
Sondernachrichten, wie z. B. Eingabeaufforderungen und tellop-Befehle

LOG_INFO
Informationsnachrichten, wie z. B. Jobstarts sowie Statusänderungen von Jobs und Jobnetzen

Durch das Setzen der Option **syslog local** auf einen positiven Wert wird die von Tivoli Workload Scheduler verwendete syslog-Funktion definiert. Das Setzen des Werts **4** bewirkt z. B., dass Tivoli Workload Scheduler die lokale Funktion LOCAL4 verwendet. Anschließend müssen Sie die entsprechenden Einträge in der Datei **/etc/syslog.conf** vornehmen und den Dämon **syslog** rekonfigurieren. Damit LOCAL4 verwendet wird und die Tivoli Workload Scheduler-Nachrichten an die Systemkonsole gesendet werden, geben Sie die folgende Zeile in die Datei **/etc/syslog.conf** ein:

```
local4    /dev/console
```

Damit die Tivoli Workload Scheduler-Fehlernachrichten an die Benutzer **maestro** und **root** gesendet werden, geben Sie Folgendes ein:

```
local4.err    maestro,root
```

Beachten Sie, dass die Selektor- und Aktionsfelder durch mindestens einen Tabulator getrennt sein müssen. Nach der Änderung der Datei **/etc/syslog.conf** können Sie den Dämon **syslog** durch die Eingabe des folgenden Befehls rekonfigurieren:

```
kill -HUP `cat /etc/syslog.pid`
```

Der Befehl ″console″

Sie können mit dem Befehl **console** von Console Manager die Tivoli Workload Scheduler-Nachrichtenebene festlegen und die Nachrichten an Ihr Terminal leiten. Die Einstellung der Nachrichtenebene wirkt sich nur auf Batchman- und Mailman-Nachrichten aus, die zahlenmäßig überwiegen. Sie legt ebenfalls die Ebene der Nachrichten fest, die in die Standardlistdatei(en) geschrieben und an den Dämon **syslog** gesendet werden. Der folgende Befehl legt z. B. die Ebene der Batchman- und Mailman-Nachrichten auf **2** fest und sendet die Nachrichten an Ihren Computer.

```
console sess;level=2
```

Nachrichten werden solange an Ihren Computer gesendet, bis Sie entweder einen weiteren Befehl **console** ausführen oder Conman beenden. Wenn Sie das Senden von Nachrichten an Ihr Terminal stoppen wollen, können Sie den folgenden Conman-Befehl eingeben:

```
console sys
```

Den Produktionszyklus automatisieren

Die Vor- und Nachbereitung kann vollständig automatisiert werden. Dazu muss das von Tivoli zur Verfügung gestellte Jobnetz **final** bzw. eine benutzerdefinierte Entsprechung zusammen mit anderen Jobnetzen der Tivoli Workload Scheduler-Datenbank hinzugefügt werden. Eine Kopie des von Tivoli zur Verfügung gestellten Jobnetzes befindet sich im Verzeichnis *TWShome*/**Sfinal**; eine Kopie des Jobskripts befindet sich unter *TWShome*/**Jnextday**. Unter Umständen trägt ein Ausdruck dieser Kopien zum Verständnis der für die Umstellung auf einen neuen Tag erforderlichen Vorgänge bei.

Das Jobnetz **final** wird jeden Tag in den Produktionsplan eingefügt. Dadurch wird vor dem Start eines neuen Tages der Job **Jnextday** ausgeführt. Dieser Job übernimmt folgende Tasks:

1. Er stellt die Verbindungen zu allen Workstations her, damit der Hauptdomänenmanager über die neuesten Zeitplanungsinformationen verfügt.
2. Er führt den Befehl **schedulr** aus, mit dem Jobnetze für den Produktionsplan des neuen Tages ausgewählt werden.
3. Er führt den Befehl **compiler** aus, mit dem der Produktionsplan kompiliert wird.
4. Er führt den Befehl **reptr** aus, mit dem die Vorbereitungsberichte gedruckt werden.
5. Er stoppt Tivoli Workload Scheduler.
6. Er führt den Befehl **stageman** aus, um nicht abgeschlossene Jobnetze weiterzuleiten, den alten Produktionsplan zu protokollieren und den neuen Plan zu installieren.
7. Er startet Tivoli Workload Scheduler für den neuen Tag.
8. Er führt die Befehle **reptr** und **rep8** aus, mit denen die Nachbereitungsberichte für den vorherigen Tag gedruckt werden.
9. Er führt den Befehl **logman** aus, mit dem statistische Daten zu Jobs des vorherigen Tages protokolliert werden.

In den Tivoli Workload Scheduler-Dokumentationen beziehen sich die Begriffe **final** und **Jnextday** sowohl auf die von Tivoli zur Verfügung gestellten Versionen als auch auf die benutzerdefinierten Entsprechungen.

Das Jobnetz ″final″ anpassen

Sie können das Jobnetz **final** vor seiner Verwendung Ihren Anforderungen entsprechend ändern oder stattdessen ein von Ihnen erstelltes Jobnetz verwenden.

Eigene Jobnetze sollten dem von Tivoli gelieferten Jobnetz entsprechen. Dabei sollten Sie Folgendes berücksichtigen:

- Wenn Sie die Art, wie **stageman** Protokolldateinamen generiert, ändern, denken Sie daran, dass **reptr** und **logman** dieselben Namen verwenden müssen.
- Möchten Sie die Vorbereitungsberichte vor dem Start eines neuen Tages drucken, können Sie den Job **Jnextday** in zwei Jobs unterteilen. Der erste Job führt die Befehle **schedulr**, **compiler** und **reptr** aus. Der zweite Job stoppt Tivoli Workload Scheduler, führt den Befehl **stageman** aus, startet Tivoli Workload Scheduler und führt die Befehle **reptr** und **logman** aus. Anschließend kann die Ausführung des ersten Jobs für einen beliebigen Zeitpunkt vor Tagesende angesetzt werden, während die Ausführung des zweiten Jobs kurz vor Tagesende angesetzt werden kann.

Das Jobnetz ″final″ hinzufügen

Wenn Sie unter UNIX arbeiten und nach den Anweisungen in „Einen Hauptdomänenmanager nach der Installation konfigurieren” auf Seite 38 vorgegangen sind oder wenn Sie Tivoli Workload Scheduler über das Programm **Setup** in NT installiert haben, ist das Jobnetz **final** bereits in die Datenbank aufgenommen worden. Befolgen Sie andernfalls die unten aufgeführten Anweisungen, um das Jobnetz **final** oder ein entsprechendes benutzerdefiniertes Jobnetz hinzuzufügen.

1. Melden Sie sich als Benutzer **TWS** oder **maestro** am Hauptdomänenmanager an.
2. Geben Sie unter UNIX oder Windows NT an der Eingabeaufforderung folgenden Befehl ein:

   ```
   composer "add Sfinal"
   ```

   Verwenden Sie anstelle von **Sfinal** den Namen Ihres eigenen Jobnetzes, um dieses hinzuzufügen.

Einen Produktionszyklus starten

Gehen Sie wie folgt vor, um zum ersten Mal einen Produktionstag zu starten oder einen neuen Produktionstag mit einer geänderten Tagesstartzeit zu starten:

1. Melden Sie sich als Benutzer **maestro** am Hauptdomänenmanager an.
2. Führen Sie an einer Eingabeaufforderung Folgendes aus: `conman "release final"` .
   Dadurch werden die Vorbereitungsprozesse ausgeführt und die Tivoli Workload Scheduler-Produktionsprozesse gestartet.

Die Produktionsumgebung verwalten

Dieser Abschnitt bietet Ihnen Informationen zum Ändern der Tagesstartzeit für Tivoli Workload Scheduler und zum Erstellen eines Plans, um die Verarbeitung zukünftiger oder vergangener Tage zu verarbeiten.

Die Tagesstartzeit auswählen

Für den Start eines Produktionstages gibt es drei übliche Zeitpunkte:

- am frühen Morgen
- am späten Nachmittag
- um Mitternacht

Im Folgenden werden einige Aspekte der Zeitplanung beschrieben:

Startzeiten und Termine

Die angegebenen Startzeiten (Schlüsselwort *AT*) stehen immer in Beziehung zur Startzeit des Tivoli Workload Scheduler-Produktionstages. Bei Jobnetzen, deren Jobs eine Verarbeitungszeit haben, die über einen Produktionstag hinausgeht, müssen Sie möglicherweise + 1 Tag hinzufügen. Vergewissern Sie sich außerdem, dass der Termin (Schlüsselwort *UNTIL*) nach der Startzeit *m* liegt.

Schlüsselwort ’On’

Produktions- und Kalendertage müssen nicht übereinstimmen. Beginnt Ihr Produktionstag um 6:00 Uhr (Standardeinstellung), ist 5:59 Uhr die letzte Minute des Produktionstages. Daher wird ein Job, dessen Ausführung für Montag (ON MONDAY) um 5:30 Uhr definiert ist, am Montag ausgewählt und am Kalendertag Dienstag um 5:30 Uhr ausgeführt.

Schlüsselwort 'Carryforward'

Wenn Sie die Tagesstartzeit gegen Mitternacht festlegen, damit es eine Übereinstimmung mit den Kalendertagen gibt, führt das wahrscheinlich zu einer großen Anzahl weitergeleiteter Jobnetze. Dadurch wird das Verwalten des Rechenzentrums möglicherweise aufwendiger.

Die Tagesstartzeit ändern

Die Tivoli Workload Scheduler-Tagesstartzeit bezeichnet den Moment, an dem das Jobnetz **final** ausgeführt wird und die Tivoli Workload Scheduler-Prozesse gestoppt und erneut gestartet werden. Gehen Sie wie folgt vor, um die Tivoli Workload Scheduler-Tagesstartzeit anzugeben:

1. Ändern Sie die Option **start** in der Datei 'Globalopts'. Hierbei handelt es sich um die Startzeit eines Tivoli Workload Scheduler-Verarbeitungstages im 24-Stundenformat: *hhmm* (**0000**-**2359**). Die Standardstartzeit ist 6:00 Uhr.
2. Ändern Sie die Startzeit (Schlüsselwort *AT*) des Jobnetzes **final**, so dass es eine Minute vor Tagesende ausgeführt wird.

So setzen Sie die Startzeit des Produktionstages auf Mitternacht:

1. Setzen Sie die Startzeit des Jobnetzes **final** auf Mitternacht.
2. Setzen Sie die Option **start** in der Datei 'Globalopts' auf '0001'.

Wenn Sie jedoch die Option **start** auf 0000 setzen und **Jnextday** auf 2359, werden möglicherweise Zeitpläne oder Jobnetze für den soeben beendeten Tag ausgewählt, da der Befehl **schedulr** das Systemdatum verwendet und in einem kleinen Netzwerk immer die Möglichkeit besteht, dass der Befehl **schedulr** vor Mitternacht ausgeführt wird.

Einen Plan für zukünftige oder vergangene Tage erstellen

Sie können einen Plan erstellen, mit dem eine Verarbeitung durchgeführt wird, die eigentlich für in der Zukunft liegende bzw. vergangene Tage terminiert war. Mit dieser Prozedur wird ein beliebiger Verarbeitungstag praktisch erneut erstellt. Sie müssen möglicherweise so vorgehen, wenn ein Verarbeitungstag auf Grund eines Notfalls ausgefallen ist.

1. Unterbrechen Sie alle Verbindungen der Workstations im Tivoli Workload Scheduler-Netzwerk, und stoppen Sie die Workstations mit den folgenden Befehlen:

   ```
   conman "unlink @!@;noask"
   conman "stop @!@;wait"
   ```

   Damit wird die gesamte Verarbeitung im Netzwerk gestoppt.
2. Führen Sie den Befehl **schedulr** mit der Datumsoption aus, um die Datei *prodsked* zu erstellen:

   ```
   schedulr -date MM/TT/JJ
   ```

   Mit der Datumsoption können Sie einen Plan erstellen, der auf einem in der Zukunft liegenden bzw. vergangenen Verarbeitungstag basiert.
3. Führen Sie den Befehl **compiler** aus, um die Datei *symnew* zu erstellen:

   ```
   compiler (-date MM/TT/JJ)
   ```

   Sie können mit dem Befehl **compiler** die Datumsoption verwenden, um das Datum des heutigen Tages bzw. des Tages, der erneut erstellt werden soll, anzugeben. Diese Option benötigen Sie möglicherweise, wenn es sich um Jobnetze mit Eingabeparametern handelt, die von einem Datum abhängen.

Der Parameter **scheddate** wird von dem Datum abgeleitet, das mit dem Befehl **compiler** angegeben wird. Wenn Sie kein Datum angeben, wird als Standardwert das Datum verwendet, das mit dem Befehl **schedulr** angegeben wurde.

4. Führen Sie Console Manager aus, um die Tivoli Workload Scheduler-Prozesse zu stoppen:
   ```
   conman stop @!@
   ```
5. Führen Sie **stageman** aus, um die neue Symphony-Datei zu erstellen:
   ```
   stageman
   ```
6. Führen Sie Console Manager aus, um die Tivoli Workload Scheduler-Prozesse zu starten:
   ```
   conman start
   ```

Die Konfigurationsskripts verwenden

Tivoli Workload Scheduler stellt zum Einrichten Ihrer Produktionsumgebung zwei Standardkonfigurationsskripts bereit: ein Skript auf globaler und eines auf lokaler Ebene.

Standardkonfigurationsskript - jobmanrc

Eine Standardkonfigurationsskriptschablone namens *maestrohome*/**config/jobmanrc** wird mit Tivoli Workload Scheduler zur Verfügung gestellt. Sie wird automatisch als *maestrohome*/jobmanrc installiert. Der Systemadministrator kann mit diesem Skript die gewünschte Umgebung einrichten, bevor die Jobs ausgeführt werden. Wenn Sie das Skript ändern wollen, nehmen Sie Ihre Änderungen an der Arbeitskopie (*maestrohome*/jobmanrc) vor, ohne die Schablonendatei zu ändern. Die Datei enthält konfigurierbare Variablen und Kommentare, die Ihnen das Prinzip verdeutlichen. In Tabelle 3 werden die Variablen für das Skript **jobmanrc** beschrieben.

Tabelle 3. Variablen von **jobmanrc**

Variablenname	Wert
UNISON_JCL	Der Pfadname für die Skriptdatei des Jobs.
UNISON_STDLIST	Der Pfadname für die Standardlistdatei des Jobs.
UNISON_EXIT	(Setzbar) Wenn diese Variable auf **yes** gesetzt ist, wird der Job sofort beendet, sobald ein Befehl einen Exit-Code ungleich null zurückgibt. Ist die Variable auf **no** gesetzt, wird die Ausführung des Jobs fortgesetzt, wenn ein Befehl einen Exit-Code ungleich null zurückgibt. Andere Einstellungen werden als **no** interpretiert.
LOCAL_RC_OK	(Setzbar) Wenn diese Variable auf **yes** gesetzt ist, wird das lokale Konfigurationsskript des Benutzers ausgeführt (sofern vorhanden) und $UNISON_JCL als erstes Argument übergeben. Der Zugriff auf diese Option kann dem Benutzer erteilt bzw. verweigert werden. Weitere Informationen finden Sie unter „Lokales Konfigurationsskript - $HOME/.jobmanrc” auf Seite 88. Ist die Variable auf **no** gesetzt, wird das Vorhandensein eines lokalen Konfigurationsskripts ignoriert und $UNISON_JCL ausgeführt. Andere Einstellungen werden als **no** interpretiert.

Tabelle 3. Variablen von **jobmanrc** *(Forts.)*

Variablenname	Wert
MAIL_ON_ABEND	(Setzbar) Wenn diese Variable auf **yes** gesetzt ist, wird eine Nachricht an die Mailbox des angemeldeten Benutzers gesendet, wenn der Job mit einem Exit-Code ungleich null beendet wird. Diese Variable kann auch auf einen oder mehrere Benutzernamen, getrennt durch Leerzeichen, gesetzt werden; eine Nachricht wird dann an jeden Benutzer gesendet. Beispiel: ″root mis sam mary″. Ist die Variable auf **no** gesetzt, werden keine Nachrichten gesendet, wenn der Job abnormal beendet wird. Nachrichten über abnormale Beendigungen haben folgendes Format: *cpu#sched.job* *jcl-datei* failed with *exit-code* Please review *standardlistdateiname*
SHELL_TYPE	(Konfigurierbar) Ist die Variable auf **standard** gesetzt, wird die erste Zeile der JCL-Datei gelesen, um die Shell zu bestimmen, die für die Jobausführung verwendet werden soll. Wenn die erste Zeile nicht mit **#!** beginnt, wird **/bin/sh** für die Ausführung des lokalen Konfigurationsskripts oder von $UNISON_JCL verwendet. Befehle werden an die Standardlistdatei des Jobs zurückgemeldet. Ist die Variable auf **user** gesetzt, wird das lokale Konfigurationsskript oder $UNISON_JCL von der Anmelde-Shell ($UNISON_SHELL) des Benutzers ausgeführt. Befehle werden an die Standardlistdatei des Jobs zurückgemeldet. Ist die Variable auf **script** gesetzt, wird das lokale Konfigurationsskript oder $UNISON_JCL direkt ausgeführt und Befehle werden nicht zurückgemeldet, es sei denn, das lokale Konfigurationsskript oder $UNISON_JCL enthält den Befehl **set -x**. Andere Einstellungen werden als **standard** interpretiert.
USE_EXEC	(Setzbar) Wenn diese Variable auf **yes** gesetzt ist, wird der Job oder das lokale Konfigurationsskript des Benutzers mit dem Befehl **exec** ausgeführt, wodurch ein zusätzlicher Prozess vermieden wird. Diese Option wird überschrieben, wenn MAIL_ON_ABEND ebenfalls auf **yes** gesetzt ist. Alle anderen Einstellungen werden als **no** interpretiert, in solch einem Fall wird der Job oder das lokale Konfigurationsskript von einem anderen Shell-Prozess ausgeführt.

Lokales Konfigurationsskript - $HOME/.jobmanrc

Mit dem lokalen Konfigurationsskript können Benutzer die gewünschte Umgebung für die Ausführung ihrer eigenen Jobs einrichten. Das Skript kann nur unter den folgenden Bedingungen ausgeführt werden:

1. Das Standardkonfigurationsskript, jobmanrc, muss installiert sein, und die Umgebungsvariable LOCAL_RC_OK muss auf **yes** gesetzt sein (siehe Tabelle 3).
2. Wenn die Datei *maestrohome*/localrc.allow vorhanden ist, muss der Name des Benutzers in der Datei erscheinen. Wenn die allow-Datei nicht vorhanden ist, darf der Name des Benutzers nicht in der Datei *maestrohome*/localrc.deny erscheinen. Ist keine dieser Dateien vorhanden, kann der Benutzer ein lokales Konfigurationsskript verwenden.
3. Das lokale Konfigurationsskript muss im Basisverzeichnis ($HOME/.jobmanrc) des Benutzers installiert sein, und es muss über eine Ausführungsberechtigung verfügen.

Wenn Sie ein lokales Konfigurationsskript verwenden wollen, muss es mindestens die Skriptdatei ($UNISON_JCL) des Jobs ausführen. Das von Tivoli bereitgestellte Standardkonfigurationsskript, jobmanrc, führt Ihr lokales Konfigurationsskript wie folgt aus:

```
$EXECIT $USE_SHELL $HOME/.jobmanrc "$UNISON_JCL" $IS_COMMAND
```

Der Wert von USE_SHELL ist auf den Wert der jobmanrc-Variablen SHELL_TYPE gesetzt (siehe Tabelle 3 auf Seite 86). IS_COMMAND ist auf **yes** gesetzt, wenn der Job mit der Anweisung **docommand** geplant oder übergeben wurde. EXECIT ist auf **exec** gesetzt, wenn die Variable USE_EXEC auf **yes** gesetzt ist (siehe Tabelle 3 auf Seite 86), andernfalls hat sie den Wert null. Im folgenden Beispiel wird gezeigt, wie die Skriptdatei eines Jobs oder ein Befehl in Ihrem lokalen Konfigurationsskript ausgeführt wird:

```
#!/bin/ksh
PATH=maestrohome:maestrohome/bin:$PATH
export PATH
/bin/sh -c "$UNISON_JCL"
```

Nachfolgend wird das Beispiel eines jobmanrc-Skripts aufgeführt, dessen Verarbeitung auf dem Exit-Code des Benutzerjobs basiert:

```
#!/bin/sh
#
PATH=maestrohome:maestrohome/bin:$PATH
export PATH
/bin/sh -c "$UNISON_JCL"
#or use eval "$UNISON_JCL" and the quotes are required
RETVAL=$?
if [ $RETVAL -eq 1 ]
then
echo "Exit code 1 - Non Fatal Error"
exit 0
elif [ $RETVAL -gt 1 -a $RETVAL -lt 100 ]
then
conman "tellop This is a database error - page the dba"
elif [ $RETVAL -ge 100 ]
then
conman "tellop Job aborted. Please page the admin"
fi
```

A

Zeitzonen und Prüfung

In diesem Anhang werden die folgenden Funktionen beschrieben:

- Zeitzonen
- Prüfung

Zeitzonen

Tivoli Workload Scheduler unterstützt Zeitzonen. Das Aktivieren von Zeitzonen bietet Ihnen die Möglichkeit, die Auslastung auf globaler Ebene zu verwalten. Die Implementierung von Zeitzonen ermöglicht außerdem eine einfache Zeitplanung bei mehreren Zeitzonen und bei Jobs, die in einem ″gesperrten Bereich″ ausgeführt werden müssen. Ein gesperrter Bereich ist der Zeitraum zwischen der Zeit, zu der der Tivoli Workload Scheduler-Tag auf dem Master beginnt, und der Zeit auf dem fehlertoleranten Agenten in einer anderen Zeitzone. Wenn z. B. ein Master, dessen Tivoli Workload Scheduler-Tag um 6:00 Uhr beginnt, einen weiter westlich gelegenen Agenten initialisiert, dessen Zeitzonenunterschied 3 Stunden beträgt, liegt der gesperrte Bereich dieses fehlertoleranten Agenten zwischen 3:00 Uhr und 6:00 Uhr. Bisher war eine Sonderverarbeitung erforderlich, um während dieses Zeitraums Jobs auszuführen. Wenn Sie jetzt eine Zeitzone für die Startzeit eines Jobs oder Jobnetzes angeben, führt Tivoli Workload Scheduler ihn/es wie erwartet aus. Nehmen wir z. B. folgende zwei Jobnetze für einen fehlertoleranten Agenten (Zeitzone PST, Pacific Standard Time) und einen Master (Zeitzone EST, Eastern Standard Time) an:

```
Schedule PST_SCHEDULE1
On SU, weekdays except FR
CARRYFORWARD
AT 0330
:
job1
job2
END
```

```
Schedule PST_SCHEDULE2
on weekdays
AT 0330 TZ PST
:
joba
jobb
END
```

Für das Jobnetz PST_SCHEDULE1 ist die Zeitzonenfunktion nicht aktiviert. Damit dieses Jobnetz am Morgen jedes Wochentages ausgeführt wird, müssen Sie es so terminieren, dass es von Sonntag bis Donnerstag ausgeführt wird. Außerdem müssen Sie die Option **CARRYFORWARD** angeben. Ohne die Option **CARRYFORWARD** würden die Jobs nie ausgeführt werden, weil der fehlertolerante Agent (unter Annahme eines EST-Masters mit dem Beginn des Tivoli Workload Scheduler-Tags um 6:00 Uhr) jeden Morgen um 3:00 Uhr initialisiert würde. Für das Jobnetz PST_SCHEDULE2 ist die Zeitzonenfunktion aktiviert. Wenn der EST-Master den fehlertoleranten Agenten (PST) um 3:00 Uhr initialisiert, wird das Jobnetz am selben Tag um 3:30 Uhr gestartet.

Das Aktivieren von Zeitzonen hat auch Auswirkungen auf fehlertolerante Agenten, wenn die Zeitplanung auf einem weiter westlich gelegenen Master durchgeführt wird. Nehmen wir z. B. folgende zwei Jobnetze für einen fehlertoleranten Agenten (Zeitzone EST, Eastern Standard Time) und einen Master (Zeitzone PST, Pacific Standard Time) an, dessen Tag um 6:00 Uhr beginnt:

```
Schedule EST_SCHEDULE1
On SU, weekdays except FR
AT 0800 + 1 DAY
CARRYFORWARD
:
job1
job2
END
```

```
Schedule EST_SCHEDULE2
On SU, weekdays except FR
AT 0800 TZ EST
:
joba
jobb
END
```

Für das Jobnetz EST_SCHEDULE1 ist die Zeitzonenfunktion nicht aktiviert. Damit dieses Jobnetz am Morgen jedes Wochentages ausgeführt wird, müssen Sie es so terminieren, dass es von Sonntag bis Donnerstag ausgeführt wird. Geben Sie die Option **CARRYFORWARD** und +1DAY für die AT-Zeit an. Die Option **CARRYFORWARD** ist für die Spezifikation +1DAY erforderlich. Ohne die Spezifikation +1DAY würde das Jobnetz sofort nach der Initialisierung durch den westlichen Master um 9:00 Uhr gestartet werden. Für das Jobnetz EST-_SCHEDULE2 ist die Zeitzonenfunktion aktiviert. Wenn der fehlertolerante Agent (EST) um 9:00 Uhr initialisiert wird, wird das Jobnetz am nächsten Tag um 8:00 Uhr ausgeführt.

Nachdem die Zeitzonenfunktion aktiviert wurde, können Sie in Job Scheduling Console oder in dem Erstellungsprogramm für Startzeiten und Termine von Jobs und Jobnetzen angegeben werden. In Conman akzeptieren die folgenden Befehle jetzt Zeitzonenparameter, wenn AT- oder UNTIL-Zeiten verwendet werden:

- submit job
- submit docommand
- submit file
- submit schedule
- addep schedule
- addep job
- addep schedule
- addep job
- Option ″rurun job; from″

Die Zeitzonenfunktion aktivieren

Die Zeitzonenfunktion wird durch einen Eintrag in der Datei **globalopts** aktiviert und indem in der Master-Workstationdefinition eine Zeitzone angegeben wird. Geben Sie dazu Folgendes an:

```
timezone enable = yes|no
```

Nach der Installation oder Aktualisierung des Produktes sind Zeitzonen standardmäßig inaktiviert. Wenn der Eintrag **timezone enable** in der Datei **globalopts** fehlt, ist die Zeitzonenoption inaktiviert.

Namen und Beschreibungen der Zeitzonen

In der folgenden Tabelle werden die durch Tivoli Workload Scheduler unterstützten Zeitzonennamen angezeigt:

Name	Bedeutung	Relativ zur WEZ
GMT	Greenwich Mean Time (Westeuropäische Zeit - WEZ)	WEZ
UTC	Universal Coordinated Time (Westeuropäische Zeit - WEZ)	WEZ
ECT	European Central Time (Mitteleuropäische Zeit - MEZ)	WEZ+1:00
EET	Eastern European Time (Osteuropäische Zeit - OEZ)	WEZ+2:00
ART	(Arabic) Egypt Standard Time	WEZ+2:00
EAT	Eastern African Time	WEZ+3:00
MET	Middle East Time	WEZ+3:30
NET	Near East Time	WEZ+4:00
PLT	Pakistan Lahore Time	WEZ+5:00
IST	India Standard Time	WEZ+5:30
BST	Bangladesh Standard Time	WEZ+6:00
VST	Vietnam Standard Time	WEZ+7:00
CTT	China Taiwan Time	WEZ+8:00
JST	Japan Standard Time	WEZ+9:00
ACT	Australia Central Time	WEZ+9:30
AET	Australia Eastern Time	WEZ+10:00
SST	Solomon Standard Time	WEZ+11:00
NST	New Zealand Standard Time	WEZ+12:00
MIT	Midway Islands Time	WEZ-11:00
HST	Hawaii Standard Time	WEZ-10:00
AST	Alaska Standard Time	WEZ-9:00
PST	Pacific Standard Time	WEZ-8:00
PNT	Phoenix Standard Time	WEZ-7:00
MST	Mountain Standard Time	WEZ-7:00
CST	Central Standard Time	WEZ-6:00
EST	Eastern Standard Time	WEZ-5:00
IET	Indiana Eastern Standard Time	WEZ-5:00
PRT	Puerto Rico and US Virgin Islands Time	WEZ-4:00
CNT	Canada Newfoundland Time	WEZ-3:30
AGT	Argentina Standard Time	WEZ-3:00
BET	Brazil Eastern Time	WEZ-3:00
CAT	Central African Time	WEZ-1:00

Prüfung

Eine Prüfoption wurde hinzugefügt, mit deren Hilfe Änderungen in der Datenbank und im Plan verfolgt werden können:

- Bei der Datenbank werden alle von Benutzern vorgenommenen Änderungen protokolliert. Das Delta der Änderungen bzw. das Vor- und Nachimage werden jedoch nicht aufgezeichnet. Das Öffnen und Sichern eines Objektes wird protokolliert, selbst wenn keine Änderungen vorgenommen wurden.
- Beim Plan werden alle vom Benutzer vorgenommenen Änderungen protokolliert. Alle Vorgänge werden unabhängig von ihrem Ergebnis (erfolgreich oder nicht) aufgezeichnet.

Die Prüfprotokolle werden in den folgenden Verzeichnissen erstellt:

```
TWShome/audit/plan
TWShome/audit/database
```

Prüflistendateien werden in einer unstrukturierten Textdatei auf einzelnen Maschinen im Tivoli Workload Scheduler-Netzwerk protokolliert. Dadurch wird das Risiko eines Prüffehlers aufgrund von Netzwerkproblemen minimiert und eine direkte Methode zum Schreiben der Protokolle aktiviert. Die Protokollformate für den Plan und die Datenbank sind weitgehend identisch. Die Protokolle umfassen einen Header, der für alle Datensätze derselbe ist, eine Aktions-ID und einen Abschnitt mit Daten, dessen Inhalt von der Aktionsart abhängt. Alle Daten sind in Klartext und so formatiert, dass sie mit einem Texteditor wie ’VI’ oder ’Notepad’ gelesen und editiert werden können.

Anmerkung: Bei Änderungsbefehlen werden für Ressourcen, Kalender, Parameter und Eingabeaufforderungen zwei Einträge in das Protokoll eingetragen. Der Änderungsbefehl wird im Protokoll als Löschbefehl und Befehl zum Hinzufügen angezeigt.

Die Prüffunktion aktivieren

Die Prüfoption wird durch zwei Einträge in der Datei **globalopts** aktiviert:

```
plan audit level = 0|1
database audit level = 0|1
```

Der Wert **1** aktiviert die Prüfung, der Wert **0** inaktiviert die Prüfung. Die Standardeinstellung in Tivoli Workload Scheduler ist zurzeit **0**, d. h., die Prüfung ist inaktiviert. Wenn diese Optionen in der Datei **globalopts** nicht vorhanden sind, ist die Prüfung inaktiviert. Bei der Installation des Produktes ist die Prüfung standardmäßig inaktiviert.

Zum Einleiten der Datenbankprüfung müssen Sie Tivoli Workload Scheduler vollständig herunterfahren und den Befehl **wmaeutil** zum Stoppen der Connector-Instanz verwenden. Wenn Sie Tivoli Workload Scheduler und die Connector-Instanz erneut starten, wird das Datenbankprüfprotokoll initialisiert. Die Planprüfung findet statt, wenn Jnextday ausgeführt wird.

Das Format des Prüfprotokolls

Das Format der Prüfprotokolle ist für den Plan und die Datenbank grundsätzlich dasselbe. Das Protokoll umfasst einen Header, eine Aktions-ID und Abschnitte mit Daten, deren Inhalt von der Aktionsart abhängt. Die Daten haben Klartextformat, und die Datenelemente sind durch einen vertikalen Balken (|) getrennt.

Die Protokolldateieinträge haben das folgende Format:

```
Protokollart|Datum (WEZ)|Zeit (WEZ)|Lokales Datum|Lokale
Zeit|Objektart|Aktionsart|Workstationname|
Benutzer-ID|Framework-Benutzer|Objektname|
Aktionsdatenfelder
```

Die Protokolldateien umfassen die folgenden Informationen:

Protokollart
In diesem Feld wird ein Wert aus acht Zeichen angezeigt, der die Quelle des Protokollsatzes angibt. Die folgenden Protokollarten werden unterstützt:

HEADER
Header der Protokolldatei

CONMAN
Text des **conman**-Befehls

FILEAID
Befehl, der eine Datei öffnet

PLAN Planaktion

STAGEMAN
stageman-Ausführung

RELEASE
Text des **release**-Befehls

DATABASE
Datenbankaktion

PARMS
Parameterbefehlstext

MAKESEC
makesec-Ausführung

DBEXPAND
dbexpand-Ausführung

Datum (WEZ)
In diesem Feld wird das Datum (WEZ) angezeigt, an dem die Aktion ausgeführt wurde. Das Format ist *jjjjmmtt*, wobei *jjjj* für das Jahr, *mm* für den Monat und *tt* für den Tag steht.

Zeit (WEZ)
In diesem Feld wird die Zeit (WEZ) angezeigt, zu der die Aktion ausgeführt wurde. Das Format ist *hhmmss*, wobei *hh* für die Stunde, *mm* für die Minute und *ss* für die Sekunde steht.

Lokales Datum
In diesem Feld wird das lokale Datum angezeigt, an dem die Aktion ausgeführt wurde. Das lokale Datum wird durch die Zeitzonenoption der Workstation definiert. Das Format ist *jjjjmmtt*, wobei *jjjj* für das Jahr, *mm* für den Monat und *tt* für den Tag steht.

Lokale Zeit
In diesem Feld wird die lokale Zeit angezeigt, zu der die Aktion ausgeführt wurde. Die lokale Zeit wird durch die Zeitzonenoption der Workstation definiert. Das Format ist *hhmmss*, wobei *hh* für die Stunde, *mm* für die Minute und *ss* für die Sekunde steht.

Objektart
In diesem Feld wird die Art des Objekts angezeigt, für das eine Aktion ausgeführt wurde. Es gibt folgende Objektarten:

DATABASE
Definition einer Datenbank

DBWKSTN
Definition einer Datenbankworkstation

DBWKCLS
Definition einer Datenbankworkstationklasse

DBDOMAIN
Definition einer Datenbankdomäne

DBUSER
Definition eines Datenbankbenutzers

DBJBSTRM
Definition eines Datenbankjobnetzes

DBJOB
Definition eines Datenbankjobs

DBCAL
Definition eines Datenbankkalenders

DBPROMPT
Definition einer Datenbankeingabeaufforderung

DBPARM
Definition eines Datenbankparameters

DBRES
Definition einer Datenbankressource

DBSEC
Datenbanksicherheit

PLAN Plan

PLWKSTN
Planworkstation

PLDOMAIN
Plandomäne

PLJBSTRM
Planjobnetz

PLJOB
Planjob

PLPROMPT
Planeingabeaufforderung

PLRES
Planressource

PLFILE
Plandatei

Aktionsart
In diesem Feld wird angezeigt, welche Aktion für das Objekt ausgeführt wurde. Die entsprechenden Werte für dieses Feld hängen von der ausgeführten Aktion ab.

Für die Datenbank kann die Aktionsart ADD, DELETE, MODIFY, EXPAND oder INSTALL sein. Tivoli Workload Scheduler zeichnet die Aktionen ADD, DELETE und MODIFY für Workstations, Workstationklassen, Domänen, Benutzer, Jobs, Jobnetze, Kalender, Eingabeaufforderungen, Ressourcen und Parameter in der Datenbank auf. Im Feld **Aktionsart** wird außerdem die Installation einer neuen Sicherheitsdatei aufgezeichnet. Wenn der Befehl **makesec** ausgeführt wird, zeichnet Tivoli Workload Scheduler ihn als INSTALL-Aktion eines Sicherheitsdefinitionsobjekts auf. Wenn der Befehl **dbexpand** ausgeführt wird, wird er als EXPAND-Aktion für ein Objekt DATABASE aufgezeichnet. LIST- und DISPLAY-Aktionen für Objekte werden nicht protokolliert. Für **fileaid** protokolliert Tivoli Workload Scheduler nur die Befehle, deren Ergebnis das Öffnen einer Datei ist. Für Parameter werden die Argumente aus der Befehlszeile protokolliert.

Workstationname
In diesem Feld wird die Tivoli Workload Scheduler-Workstation angezeigt, von der aus der Benutzer die Aktion ausführt.

Benutzer-ID
In diesem Feld wird der angemeldete Benutzer angezeigt, der die betreffende Aktion ausgeführt hat. Auf Win32-Plattformen ist dies der vollständig qualifizierte Domänenname *domäne\benutzer*.

Framework-Benutzer
In diesem Feld wird die von Tivoli Management Framework erkannte Benutzer-ID angezeigt. Dies ist die Anmelde-ID des Job Scheduling Console-Benutzers.

Objektname
In diesem Feld wird der vollständig qualifizierte Name des Objektes angezeigt. Das Format des Feldes hängt von der in der folgenden Liste aufgelisteten Objektart ab:

DATABASE
Nicht zutreffend

DBWKSTN
workstation

DBWKCLS
workstationklasse

DBDOMAIN
domäne

DBUSER
[*workstation#*]*benutzer*

DBJBSTRM
workstation#jobnetz

DBJOB
workstation#job

DBCAL
kalender

DBPROMPT
eingabeaufforderung

DBPARM
workstation#parameter

DBRES
workstation#ressource

DBSEC
Nicht zutreffend

PLAN Nicht zutreffend

PLWKSTN
workstation

PLDOMAIN
domäne

PLJBSTRM
workstation#jobnetz-instanz

PLJOB
workstation#jobnetz-instanz.job

PLPROMPT
[*workstation#*]*eingabeaufforderung*

PLRES
workstation#*ressource*

PLFILE
workstation#*pfad*(*qualifikationsmerkmal*)

Aktionsdatenfelder
In diesem Feld werden die aktionsspezifischen Datenfelder angezeigt. Das Format dieser Daten hängt vom Feld **Aktionsart** ab.

Prüfprotokoll-Header

Jede Protokolldatei beginnt mit einem Kopfsatz, der Informationen enthält, wann das Protokoll erstellt wurde und ob es ein Plan- oder Datenbankprotokoll ist.

Der Header-Dateieintrag hat den folgenden Inhalt:

Protokollart
HEADER

Datum (WEZ)
Das Datum (WEZ), an dem die Protokolldatei erstellt worden ist

Zeit (WEZ)
Die Zeit (WEZ), zu der die Protokolldatei erstellt worden ist

Lokales Datum
Das lokale Datum, an dem die Protokolldatei erstellt worden ist

Lokale Zeit
Die lokale Zeit, zu der die Protokolldatei erstellt worden ist

Workstationname
Der Tivoli Workload Scheduler-Workstationname, für den diese Datei erstellt worden ist. Jede Workstation im Tivoli Workload Scheduler-Netzwerk erstellt ihr eigenes Protokoll.

Benutzer-ID
Die Tivoli Workload Scheduler-Benutzer-ID, die die Protokolldatei erstellt hat

Objektart
Bei einer Datenbankprotokolldatei steht in diesem Feld DATABASE, bei einer Planprotokolldatei steht in dem Feld PLAN.

Objektname
Nicht zutreffend

Aktionsart
Nicht zutreffend

Aktionsdatenfelder
In diesem Feld wird die Version der Datei angezeigt.

Beispieleinträge für Prüfprotokolle

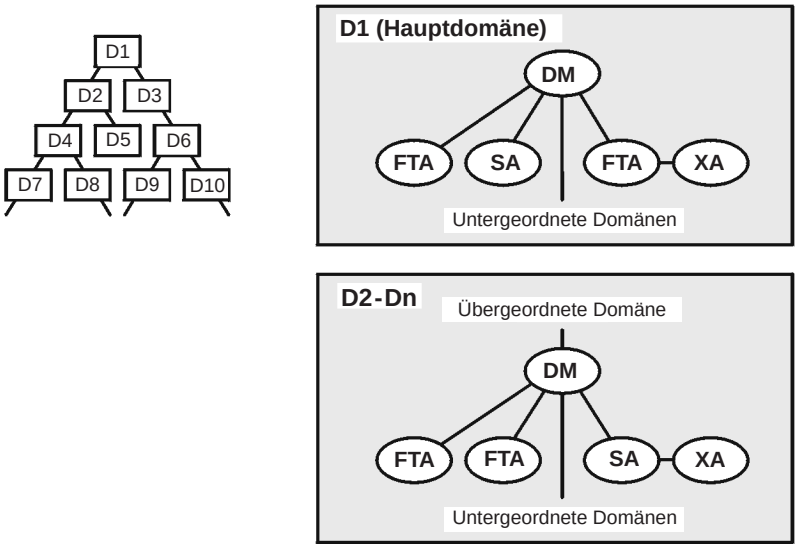
Es folgen einige Beispieleinträge für eine Protokolldatei:

```
HEADER  |19991202|201200|19991202|131200|DATABASE|        |GANGES
|RIVERS\pyasa                |||1.0
DATABASE|19991202|224504|19991202|154504|DBWKSTN |ADD
|GANGES |RIVERS\pyasa         ||JAMUNA|
DATABASE|19991203|001400|19991202|171400|DBJOB   |MODIFY  |SINDHU
|RIVERS\tairak               ||NARMADA#dubo|
```

B

Tivoli Workload Scheduler-Netzwerke

Ein Tivoli Workload Scheduler-Netzwerk umfasst eine oder mehrere Domänen, die hierarchisch angeordnet sind. Eine Tivoli Workload Scheduler-Domäne ist eine logische Gruppierung von Computern, die einen Domänenmanager und eine Anzahl Agenten umfasst.



Definitionen

Sicherungsdomänenmanager
: Ein fehlertoleranter Agent, der die Rolle seines Domänenmanagers übernehmen kann.

Domäne
: Eine benannte Gruppe von Tivoli Workload Scheduler-Workstations, die aus einem oder mehreren Agenten und einem Domänenmanager besteht. Alle Domänen haben eine übergeordnete Domäne.

Domänenmanager (DM)
: Der Management-Hub in einer Domäne. Die Kommunikation der Agenten in einer Domäne wird vollständig über den Domänenmanager geleitet. Siehe auch “Hauptdomänenmanager”.

Erweiterter Agent (Extended Agent, XA)
Eine Agentenworkstation, die Jobs nur unter der Steuerung des zugehörigen Hosts startet. Erweiterte Agenten können als Schnittstelle zwischen Tivoli Workload Scheduler und Systemen und Anwendungen verwendet werden, die nicht zu Tivoli Workload Scheduler gehören.

Fehlertoleranter Agent (FTA)
Eine Agentenworkstation, die lokale Abhängigkeiten auflösen und zugehörige Jobs ohne Domänenmanager starten kann

Host (X-Host)
Die Planungsfunktion, die für erweiterte Agenten erforderlich ist. Sie kann von einer beliebigen Tivoli Workload Scheduler-Workstation mit Ausnahme eines anderen erweiterten Agenten ausgeführt werden.

Hauptdomänenmanager
Der Domänenmanager der obersten Domäne eines Tivoli Workload Scheduler-Netzwerks. Er enthält die zentralen Master-Dateien, die zur Dokumentierung von Zeitplanungsobjekten verwendet werden. Darüber hinaus erstellt er bei jedem Tagesstart die Produktionssteuerungsdatei und führt sämtliche Protokollierungs- und Berichtsfunktionen für das Netzwerk aus. Siehe auch “Domänenmanager”.

Hauptdomäne
Die oberste Domäne in einem Maestro-Netzwerk

Übergeordnete Domäne
Die Domäne, die sich direkt über der aktuellen Domäne befindet. Alle Domänen mit Ausnahme der Hauptdomäne haben eine übergeordnete Domäne. Die Kommunikation einer Domäne wird über den übergeordneten Domänenmanager geleitet.

Standardagent (SA)
Eine Agenten-CPU, die Jobs nur unter der Steuerung des zugehörigen Domänenmanagers startet

Tivoli Workload Scheduler for MPE

Tivoli Workload Scheduler-Netzwerke können eine Kombination von MPE (ein Hewlett-Packard-eigenes Betriebssystem), Windows NT, UNIX und anderen Computern und Agenten umfassen. Weitere Informationen finden Sie im Handbuch *Maestro for MPE User's Guide*.

Netzwerkkommunikation

In einem Tivoli Workload Scheduler-Netzwerk kommunizieren die Agenten mit ihren jeweiligen Domänenmanagern, und diese wiederum mit ihren übergeordneten Domänenmanagern. Grundsätzlich werden zwei Arten von Kommunikation ausgeführt: 1) Initialisierung der Tagesstartzeit und 2) Planungsereignisse in Form von Statusänderungsnachrichten während des Verarbeitungstages.

Der Hauptdomänenmanager erstellt vor dem Start eines Produktionstages eine Produktionssteuerungsdatei, die so genannte Symphony-Datei. Anschließend wird innerhalb des Netzwerkes ein Neustart von Tivoli Workload Scheduler durchgeführt, und der Hauptdomänenmanager sendet eine Kopie der neuen Symphony-Datei an jeden Agenten, der automatisch mit ihm verbunden wird, sowie an untergeordnete Domänenmanager.

Die Domänenmanager senden ihrerseits Kopien an die Agenten, die automatisch mit ihnen verbunden werden, sowie an untergeordnete Domänenmanager. Agenten und Domänenmanager, die nicht für eine automatische Verbindung eingerichtet sind, werden mit einer Kopie der Symphony-Datei initialisiert, sobald eine Verbindungsoperation in Tivoli Workload Scheduler ausgeführt worden ist.

Wenn das Netzwerk gestartet wurde, werden Planungsnachrichten, wie z. B. Nachrichten zum Jobstart und -ende, von den Agenten an die jeweiligen Domänenmanager übergeben und anschließend über die übergeordneten Domänenmanager an den Hauptdomänenmanager übermittelt. Der Hauptdomänenmanager verteilt diese Nachrichten dann über die hierarchische Baumstruktur, um die Symphony-Dateien aller Domänenmanager und fehlertoleranten Agenten im Modus 'Vollständiger Status' zu aktualisieren.

Netzwerkverbindungen

Verbindungen ermöglichen die bidirektionale Kommunikation zwischen Tivoli Workload Scheduler-Workstations in einem Netzwerk. Verbindungen werden durch die Markierung **Autolink** und durch die Console Manager-Befehle **Link** und **Unlink** gesteuert. Wenn eine Verbindung geöffnet ist, werden Nachrichten zwischen zwei Workstations übergeben. Wenn eine Verbindung geschlossen ist, speichert die sendende Workstation Nachrichten in einer lokalen pobox-Datei und sendet sie an die Zielworkstation, wenn die Verbindung wieder geöffnet wird.

Anmerkung: Erweiterte Agenten haben keine Verbindungen. Sie kommunizieren mit ihren Domänenmanagern über ihre Hosts.

Aktivieren Sie die Markierung **Autolink** in der Definition der Workstation, damit eine Workstationverbindung automatisch geöffnet wird. Die Verbindung wird zum ersten Mal geöffnet, wenn Tivoli Workload Scheduler auf der Hauptdomänenworkstation gestartet wird. Wenn der Domänenmanager und die Workstations der untergeordneten Domäne nicht initialisiert sind und ihre Markierung **Autolink** aktiviert ist, versucht der Hauptdomänenmanager, eine Verbindung zu seinen untergeordneten Elementen herzustellen und den Initialisierungsprozess zu beginnen. Wenn die Markierung **Autolink** inaktiviert ist, wird die Workstation erst initialisiert, wenn der Befehl **Link** auf dem Hauptdomänenmanager ausgeführt wird. Nachdem die Workstation initialisiert ist, wird sie automatisch gestartet, und die Verbindung zu ihrem Domänenmanager wird aufgebaut.

Wenn Sie eine Workstation stoppen, werden ihre Pfade zu anderen Workstations geschlossen. Die Pfade von anderen Workstations zu dieser Workstation bleiben allerdings geöffnet, bis eines der folgenden Ereignisse eintritt:

- Die gestoppte Workstation wird erneut gestartet, und ein Befehl **Link** wird abgesetzt.
- Die **Mailman**-Prozesse der anderen Workstations überschreiten das Zeitlimit.

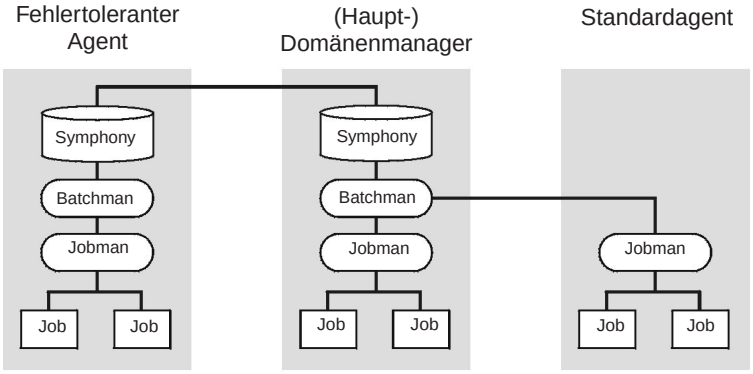
Sie können einen Befehl **Link** absetzen, nachdem Sie eine Workstation erneut gestartet haben, um sicherzustellen, dass die Kommunikation zwischen Workstations korrekt wiederhergestellt wurde.

Netzwerkoperation

Die Batchman-Prozesse auf den Domänenmanagern und den fehlertoleranten Agenten werden unabhängig ausgeführt, wobei sie die jeweilige Symphony-Datei durchsuchen, um Abhängigkeiten aufzulösen und Jobs zu starten. Batchman startet Jobs über den Jobman-Prozess. Auf einem Standardagenten antwortet der Jobman-Prozess auf Startanforderungen durch Batchman auf dem Domänenmanager.

Der Hauptdomänenmanager wird fortlaufend über Jobstarts und -beendigungen informiert und ist dafür verantwortlich, die Informationen an Domänenmanager und fehlertolerante Agenten weiterzumelden, damit sie vorhandene Abhängigkeiten zwischen Workstations auflösen können.

Der Synchronisationsgrad zwischen den Symphony-Dateien hängt von den Einstellungen der Modi 'Vollständiger Status' und 'Abhängigkeiten auflösen' in der Definition einer Workstation ab. Wenn diese Modi aktiviert sind, enthält die Symphony-Datei eines fehlertoleranten Agenten dieselben Informationen wie die Datei auf dem Hauptdomänenmanager. (Nähere Angaben finden Sie in dem Abschnitt im *Tivoli Workload Scheduler Console Benutzerhandbuch*, der beschreibt, wie Sie Workstations in der Datenbank verwalten.)

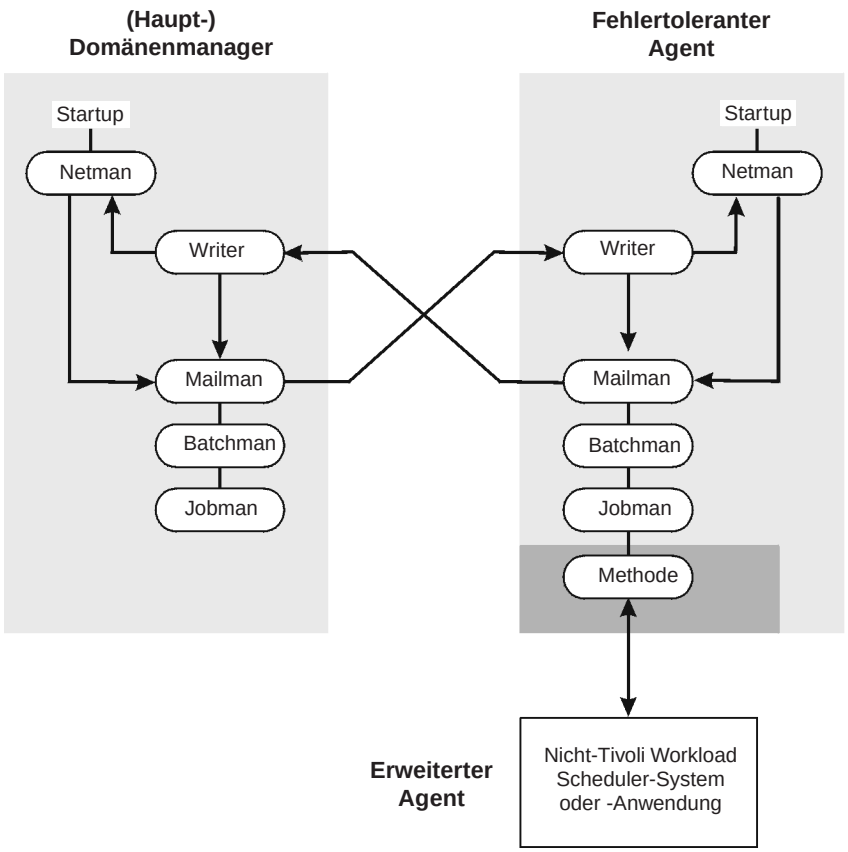


Netzwerkprozesse

Netman wird durch das Skript **StartUp** gestartet. Die Reihenfolge der Prozesserstellung ist wie folgt: Netman, Mailman, Batchman und Jobman. Auf Standardagentenworkstations wird Batchman nicht ausgeführt. Alle Prozesse mit Ausnahme von Jobman werden als Benutzer **TWS** ausgeführt. Jobman wird als **root** ausgeführt.

Wenn die Netzwerkaktivität startet, empfängt Netman Anforderungen von fernen Mailman-Prozessen. Sobald eine Anforderung empfangen wurde, erstellt Netman einen Writer-Prozess und übergibt die Verbindung an ihn. Writer empfängt die Nachricht und übergibt sie an den lokalen Mailman-Prozess. Die Writer-Prozesse (auf einem Domänenmanager können mehr als einer vorhanden sein) werden durch Verbindungsanforderungen gestartet und durch Anforderungen zum Aufheben der Verbindung gestoppt (oder wenn der kommunizierende Mailman-Prozess beendet wird).

Domänenmanager einschließlich des Hauptdomänenmanagers können mit einer großen Anzahl Agenten und untergeordneter Domänenmanager kommunizieren. Sie können Mailman-Server auf einem Domänenmanager definieren, um die Kommunikationsauslastung zur Leistungsverbesserung zu verteilen. (Nähere Angaben finden Sie in dem Abschnitt im *Tivoli Workload Scheduler Console Benutzerhandbuch*, der beschreibt, wie Sie Workstations in der Datenbank verwalten.)



Erweiterte Agenten

Ein erweiterter Agent (XA oder X-Agent) dient als Schnittstelle zu einem externen System oder einer externen Anwendung, das/die nicht zu Tivoli Workload Scheduler gehört. Es/sie wird als Tivoli Workload Scheduler-Workstation mit einer Zugriffsmethode und einem Host definiert. Die Zugriffsmethode kommuniziert mit dem externen System oder der externen Anwendung, um Jobs zu starten oder zu überwachen und die Dateiabhängigkeit **Opens** zu testen. Der Host ist eine andere Tivoli Workload Scheduler-Workstation (mit Ausnahme eines anderen erweiterten Agenten), die Abhängigkeiten auflöst und Jobstartanforderungen über die Methode absetzt.

Jobs werden für einen erweiterten Agenten in derselben Weise definiert wie für andere Tivoli Workload Scheduler-Workstations, mit der Ausnahme, dass Jobattribute durch das externe System oder die externe Anwendung vorgegeben werden.

Software für erweiterte Agenten ist für verschiedene Systeme und Anwendungen verfügbar. Die erweiterten UNIX-Agenten, die in Tivoli Workload Scheduler integriert sind, werden im folgenden Abschnitt beschrieben. Kontaktieren Sie Ihren Tivoli-Vertriebsbeauftragten, um Informationen zu anderen erweiterten Agenten zu erhalten. Informationen zur Definition von Tivoli Workload Scheduler-Workstations finden Sie in dem Abschnitt im *Tivoli Workload Scheduler Console Benutzerhandbuch*, der beschreibt, wie Sie Workstations in der Datenbank verwalten. Informationen zum Schreiben von Zugriffsmethoden finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

Erweiterte UNIX-Agenten

Tivoli Workload Scheduler umfasst Zugriffsmethoden für zwei Arten von erweiterten UNIX-Agenten. Mit der Methode für Zugriff auf lokalem UNIX-Agenten kann ein einzelner UNIX-Computer als zwei Tivoli Workload Scheduler-Workstations dienen, die beide mit Tivoli Workload Scheduler terminierte Jobs ausführen können. Mit der Methode für Zugriff auf fernem UNIX-Agenten können Sie einen fernen UNIX-Computer bestimmen, damit er mit Tivoli Workload Scheduler terminierte Jobs ausführt, ohne dass Tivoli Workload Scheduler auf ihm installiert ist.

Informationen zur Ausführung eines Jobs werden von einem erweiterten Agenten über die Datei **stdlist** des Jobs an Tivoli Workload Scheduler gesendet. In einer Optionsdatei für Methoden können alternative Anmeldenamen zum Starten von Jobs und Überprüfen von Dateiabhängigkeiten **Opens** angegeben werden. Weitere Informationen finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

Die Methode für Zugriff auf lokalem UNIX-Agenten

Die Methode für Zugriff auf lokalem UNIX-Agenten kann verwendet werden, um mehrere Tivoli Workload Scheduler-Workstations auf einem Computer zu definieren: die Hostworkstation und mindestens einen erweiterten Agenten. Wenn Tivoli Workload Scheduler einen Job an einen lokalen erweiterten UNIX-Agenten sendet, wird die Zugriffsmethode **unixlocl** durch den Host aufgerufen, um den Job auszuführen. Die Methode startet, indem sie das Standardkonfigurationsskript auf der Hostworkstation (TWShome/jobmanrc) ausführt. Wenn der angemeldete Benutzer des Jobs ein lokales Skript verwenden darf und das Skript unter dem Namen `$HOME/.jobmanrc` vorhanden ist, wird das lokale Konfigurationsskript ebenfalls ausgeführt. Der Job selbst wird dann entweder durch das Standard- oder durch das lokale Konfigurationsskript ausgeführt. Wenn keines der beiden Konfigurationsskripts vorhanden sind, startet die Methode den Job.

Das Starten der Konfigurationsskripts `jobmanrc` und `.jobmanrc` kann im Methodenskript konfiguriert werden. Die Methode führt die Konfigurationsskripts standardmäßig aus, falls sie vorhanden sind. Sie müssen eine Reihe von Zeilen in dem Methodenskript auf Kommentar setzen, um diese Funktion zu inaktivieren. Weitere Informationen finden Sie in der Skriptdatei `TWShome/methods/unixlocl` auf dem Host des erweiterten Agenten.

Die Methode für Zugriff auf fernem UNIX-Agenten

Die Methode für Zugriff auf fernem UNIX-Agenten kann verwendet werden, um festzulegen, dass auf einem nicht zu Tivoli Workload Scheduler gehörenden Computer mit Tivoli Workload Scheduler terminierte Jobs ausgeführt werden sollen. Wenn Tivoli Workload Scheduler einen Job an einen fernen erweiterten UNIX-Agenten sendet, erstellt die Zugriffsmethode `unixrsh` ein Verzeichnis `/tmp/maestro` auf dem nicht zu Tivoli Workload Scheduler gehörenden Computer. Danach überträgt sie ein Oberflächenskript in das Verzeichnis und führt es aus. Die Oberfläche führt dann den terminierten Job aus. Die Oberfläche wird nur einmal erstellt, es sei denn, sie wird gelöscht, verschoben oder ist nicht mehr aktuell.

Die angemeldeten Benutzer des Jobs müssen die erforderlichen Berechtigungen auf dem nicht zu Tivoli Workload Scheduler gehörenden UNIX-Computer erhalten, um Jobs über den erweiterten Agenten auszuführen. Dazu sollte auf dem Computer eine Datei `.rhost`, `/etc/host.equiv` oder eine entsprechende Datei eingerichtet werden. Wenn Dateiabhängigkeiten **Opens** überprüft werden sollen, muss auch die Zugriffsberechtigung **root** erteilt werden. Kontaktieren Sie Ihren Systemadministrator, falls Sie Hilfe benötigen. Weitere Informationen zu der Zugriffsmethode finden Sie in der Skriptdatei `Maestrohome/methods/unixrsh` auf dem Host eines erweiterten Agenten.

Die Produktion für erweiterte Agenten verwalten

Normalerweise verhalten sich Jobs, die auf erweiterten Agenten ausgeführt werden, wie andere Tivoli Workload Scheduler-Jobs. Tivoli Workload Scheduler überwacht den Status des Jobs und zeichnet die Ausgabe in den **stdlist**-Dateien eines Jobs auf. Diese Dateien werden auf der Hostworkstation des erweiterten Agenten gespeichert. Weitere Informationen zur Verwaltung von Jobs finden Sie in dem Abschnitt im *Tivoli Workload Scheduler Console Benutzerhandbuch*, der Tivoli Workload Scheduler-Plantasks beschreibt.

Fehler beim Starten von Jobs auf einem erweiterten Agenten

Wenn sich die Zugriffsmethode auf dem Host des erweiterten Agenten nicht im korrekten Verzeichnis befindet oder nicht mit Tivoli Workload Scheduler auf die Methode zugegriffen werden kann, schlägt das Starten von Jobs fehl, oder eine Dateiabhängigkeit wird nicht überprüft. Bei einem Job muss der Anmeldename des Tivoli Workload Scheduler-Jobs oder der in der Optionsdatei für Methoden angegebene Anmeldename Lese- und Ausführungsberechtigung für die Zugriffsmethode haben. Wenn eine Datei überprüft wird, um eine Dateiabhängigkeit **Opens** zu erfüllen, wird root als Anmeldename verwendet, wenn kein anderer Anmeldename in der Optionsdatei für Methoden angegeben ist. Weitere Informationen zu Optionen für Methoden finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*.

Die Netman-Konfigurationsdatei

Die Netman-Konfigurationsdatei ist auf allen Tivoli Workload Scheduler-Workstations vorhanden. Wenn Netman im Tivoli Workload Scheduler-Basisverzeichnis (die Standardeinstellung) installiert ist, hat die Datei den folgenden Namen: `Maestrohome/Netconf`. Wenn Netman in einem eigenen Verzeichnis installiert ist, hat die Datei den folgenden Namen: `netmanhome/Netconf`. Die Datei definiert die Dienste, die durch Netman angeboten werden. Die von Tivoli gelieferte Datei `NetConf` enthält Kommentare, die alle Dienste beschreiben. Es gibt folgende Dienste:

2001 Einen Writer-Prozess starten, der ankommende Nachrichten eines fernen Mailman-Prozesses bearbeitet

2002 Den Mailman-Prozess starten. Mailman startet dann den Rest der Prozessbaumstruktur (Batchman, Jobman).

2003 Den Tivoli Workload Scheduler-Prozess stoppen, der ankommende Nachrichten eines fernen Mailman-Prozesses bearbeitet

2004 Eine stdlist-Datei suchen und an den anfordernden Conman-Prozess zurückgeben

2005 Den Domänenmanager in einer Domäne wechseln

2501 Den Status eines fernen Jobs überprüfen

2502 Console Manager starten (ein Dienst, der von der Clientseite von Remote Console angefordert wurde)

Der Mailman-Dienst (2002) kann einen Parameter umfassen, der die Größe der internen Symphony-Tabelle festlegt. Die Tabelle sollte genug Speicherbereich für alle Datensätze in der Symphony-Datei bieten und zusätzlichen Speicherbereich für Arbeit, die übergeben worden ist, nachdem Tivoli Workload Scheduler seinen Produktionslauf gestartet hatte. Der `NetConf`-Eintrag hat die folgende Syntax:

```
2002     son     bin/mailman [-parm wert]
```

Für die Option **-parm** kann *wert* einen der folgenden Werte annehmen:

-nummer
Die Symphony-Tabelle wird mit Speicherbereich für genau diese Anzahl Datensätze erstellt. Beispiel: **-parm 6000** erstellt eine Tabelle mit Speicherbereich für genau 6000 Datensätze. Die maximal zulässige Größe ist 65.535 Datensätze. Indem Sie den Parameter auf -1 setzen, stellen Sie sicher, dass die maximale Größe verwendet wird.

nummer
Die Symphony-Tabelle wird mit Speicherbereich für alle Datensätze in der Symphony-Datei erstellt und bietet zusätzlichen Speicherbereich für diese Anzahl Datensätze.

Wenn Sie eine Nachricht empfangen, die anzeigt, dass zu viele Jobs für die Bearbeitung durch Batchman terminiert sind, ist es möglicherweise erforderlich, dass Sie die Symphony-Tabelle vergrößern. Ihr Ansprechpartner der Tivoli-Kundenunterstützung kann Ihnen beim Festlegen der angemessenen Größe helfen.

Die Netzwerk-IP-Adresse überprüfen

Wenn eine TCP/IP-Verbindung aufgebaut wird, liest Netman den Knotennamen und die IP-Adresse des Requesters vom Socket. Die IP-Adresse und der Knotenname werden verwendet, um in der Symphony-Datei nach einer bekannten Tivoli Workload Scheduler-Workstation zu suchen. Diese Suche kann folgende mögliche Ergebnisse haben:

- Wenn eine übereinstimmende IP-Adresse gefunden wurde, war die Überprüfung erfolgreich.
- Wenn ein übereinstimmender Knotenname gefunden wurde, war die Überprüfung erfolgreich.
- Wenn in der Symphony-Datei keine Übereinstimmung gefunden wurde oder die von `gethostbyname()` zurückgegebene IP-Adresse nicht mit der vom Socket gelesenen Adresse übereinstimmt, war die Überprüfung nicht erfolgreich.

Die lokale Option `nm ipvalidate` bestimmt die Aktion, die ausgeführt werden soll, wenn die Überprüfung der IP-Adresse nicht erfolgreich ist. Wenn die Option auf `full` gesetzt ist, verursacht eine nicht erfolgreiche Überprüfung, dass Tivoli Workload Scheduler die Verbindung schließt und eine Fehlernachricht generiert. Wenn die Option auf `none` gesetzt ist, lässt Tivoli Workload Scheduler alle Verbindungen zu, erstellt aber bei nicht erfolgreichen Gültigkeitsprüfungen eine Warnung.

Systemkonfiguration (nur UNIX)

Die Überprüfung einer IP-Adresse hängt davon ab, dass der Systemaufruf `gethostbyname()` alle gültigen Adressen für einen Host abruft. Das Verhalten dieser Routine ist je nach Systemkonfiguration unterschiedlich. Wenn `gethostbyname()` die Datei `/etc/hosts` verwendet, wird der erste übereinstimmende Eintrag zurückgegeben. Wenn die Verbindung mit einer Adresse eingeleitet wurde, die nach dem ersten übereinstimmenden Eintrag steht, schlägt die Überprüfung der IP-Adresse fehl. Stellen Sie den Eintrag, der verwendet wird, um die Verbindung einzuleiten, vor alle anderen übereinstimmenden Einträge in der Datei `/etc/hosts`, um dieses Problem zu lösen. Wenn `gethostbyname()` den ″benannten″ Namenserver oder den NIS-Server (Network Information Services) verwendet und `gethostbyname()` fehlschlägt, kontaktieren Sie Ihren Systemadministrator, um Unterstützung zu erhalten.

Fehlernachrichten/Warnungen

In der folgenden Liste finden Sie die Nachrichten für die Überprüfung von IP-Adressen. Wenn die lokale Option `nm ipvalidate` auf `none` gesetzt ist, werden die Fehler als Warnungen angezeigt.

- Der Tivoli Workload Scheduler-Workstationname wird in der Symphony-Datei nicht gefunden.

  ```
  IP address validation failed for request:
  Service nummer for programm on cpu(bs-art).
  Connection received from IP address:
  verb-ip-adr. MAESTRO CPU cpu not found in
  Symphony file.
  ```

- Der Aufruf von `gethostbyname()` schlägt fehl.

  ```
  IP address validation failed for request:
  Service nummer for programm on cpu(bs-art).
  Connection received from IP address:
  verb-ip-adr. gethostbyname() failed, unable to
  retrieve IP address of connecting node: knoten.
  ```

- Die von `gethostbyname()` zurückgegebenen IP-Adressen stimmen nicht mit der IP-Adresse der verbindenden Workstation überein.

  ```
  IP address validation failed for request:
  Service nummer for programm on cpu(bs-art).
  Connection received from IP address:
  verb-ip-adr. System known IP addresses for node
  name knoten: bek-ip-adr.
  ```

- Die in der Workstationdefinition angegebene IP-Adresse für die in dem Dienstanforderungspaket angegebene Tivoli Workload Scheduler-Workstation stimmt nicht mit der IP-Adresse der verbindenden Workstation überein.

  ```
  IP address validation failed for request:
  Service nummer for programm on cpu(bs-art).
  Connection received from IP address:
  verb-ip-adr. TWS known IP addresses for cpu
  bek-ip-adr.
  ```

- Unabhängig vom Status von **nm ipvalidate** wird die folgende Informationsnachricht angezeigt, wenn die Überprüfung der IP-Adresse nicht ausgeführt werden kann, weil die Symphony-Datei nicht vorhanden ist oder ein Fehler beim Lesen der Datei auftritt.

  ```
  IP address validation not performed for
  request: Service nummer for programm on
  cpu(bs-art). Connection received from IP
  address: verb-ip-adr. Cannot open or read
  Symphony file. Service request accepted.
  ```

  Dabei gilt Folgendes:

  nummer
  Dienstnummer (2001 - **Writer**, 2002 - **Mailman**, ...)

  programm
  Programm, das den Dienst anfordert

  cpu Tivoli Workload Scheduler-Workstationname der verbindenden Workstation

  bs-art Betriebssystem der verbindenden Workstation

  knoten
  Knotenname oder IP-Adresse der verbindenden Workstation

verb-ip-adr
IP-Adresse der verbindenden Workstation

bek-ip-adr
Bekannte IP-Adresse für die verbindende Workstation

Beim Fehlen einer Symphony-Datei ist die Überprüfung der IP-Adresse immer erfolgreich. In einem Tivoli Workload Scheduler-Netzwerk ist der erste Schritt zum Verbinden (Option zum automatischen Verbinden bei einem manuellen Verbindungsbefehl) eines Domänenmanager mit einem Agenten im Allgemeinen erfolgreich, da die Datei **Symphony** noch nicht vorhanden ist. Wenn auf dem Agenten allerdings eine Symphony-Datei einer früheren Ausführung vorhanden ist, schlägt die erste Verbindungsanforderung möglicherweise fehl, wenn der Name des Domänenmanagers nicht in der Symphony-Datei enthalten ist.

Fehlerbehebung im Netzwerk

Einige Problemarten erfordern möglicherweise das Ausführen von Prozeduren zur Behebung von Netzwerkfehlern. Dazu gehören folgende Probleme:

- Initialisierungsprobleme, die verhindern, dass Agenten und Domänenmanager am Beginn eines neuen Tags korrekt gestartet werden
- Netzwerkverbindungsprobleme, die verhindern, dass Agenten mit ihren Domänenmanagern kommunizieren
- Der Verlust eines Domänenmanagers, wodurch ein Wechsel auf den Sicherungsmanager erforderlich wird

Anmerkung: In allen Fällen betrifft ein Problem mit einem Domänenmanager alle seine Agenten und untergeordneten Domänenmanager.

Initialisierungsprobleme

Initialisierungsprobleme können auftreten, wenn Tivoli Workload Scheduler für einen neuen Tag gestartet wird. Die Probleme können dadurch verursacht werden, dass Tivoli Workload Scheduler-Prozesse vom vorherigen Tag oder von einer früheren Tivoli Workload Scheduler-Ausführung auf einem Agenten oder Domänenmanager ausgeführt werden. Gehen Sie wie folgt vor, um den Agenten oder Domänenmanager in dieser Situation zu initialisieren:

1. Melden Sie sich im Falle eines Domänenmanagers am übergeordneten Domänenmanager oder am Hauptdomänenmanager an. Melden Sie sich im Falle eines Agenten am Domänenmanager des Agenten, am übergeordneten Domänenmanager oder am Hauptdomänenmanager an.
2. Führen Sie Console Manager aus, und führen Sie einen Stoppbefehl für den betreffenden Agenten aus.
3. Führen Sie einen Befehl **Link** für den betreffenden Agenten aus. Dadurch wird der Agent initialisiert und gestartet.

Wenn diese Aktionen fehlschlagen, überprüfen Sie, ob Netman auf dem betreffenden Agenten ausgeführt wird. Wenn dies nicht der Fall ist, setzen Sie den Befehl **StartUp** lokal ab. Setzen Sie danach einen Befehl **Link** von seinem Domänenmanager aus ab.

Wenn schwerwiegende Netzwerkprobleme auftreten, kann ein fehlertoleranter Agent oder ein untergeordneter Domänenmanager als eigenständiges System ausgeführt werden. Stoppen Sie dazu den Agenten oder Domänenmanager, und kopieren Sie die Datei `TWShome\Sinfonia` vom Hauptdomänenmanager. Benennen Sie die kopierte Datei in `TWShome\Symphony` um, und starten Sie den Agenten oder Domänenmanager.

Alle Abhängigkeiten zwischen Workstations müssen mit entsprechenden Console Manager-Befehlen lokal aufgelöst werden, wie z. B. zum Löschen von Abhängigkeiten oder Freigeben.

Anmerkung: Damit Sie diesen Schritt ausführen können, müssen der Hauptdomänenmanager und der Agent oder untergeordnete Domänenmanager dieselbe Prozessorarchitektur haben.

Netzwerkverbindungsprobleme

Tivoli Workload Scheduler hat einen hohen Grad an Fehlertoleranz im Falle eines Kommunikationsfehlers. Jeder fehlertolerante Agent hat seine eigene Kopie der Symphony-Datei, die die Verarbeitungen dieses Tages enthält. Wenn Verbindungsfehler auftreten, führen die Agenten die Verarbeitung unter Verwendung ihrer eigenen Kopie der Symphony-Datei fort. Alle Abhängigkeiten zwischen Workstations müssen allerdings mit entsprechenden Console Manager-Befehlen lokal aufgelöst werden, wie z. B. zum Löschen von Abhängigkeiten oder Freigeben.

Solange eine Verbindung inaktiv ist, werden alle Nachrichten, die für diese nicht kommunizierende Workstation bestimmt sind, in einer Datei mit dem Namen `workstationname.msg` im Verzeichnis `TWShome\pobox` gespeichert. Wenn die Verbindung wiederhergestellt wird, fangen die Workstations an, ihre gespeicherten Nachrichten zu senden. Wenn die Verbindungen zu einem Domänenmanager für einen längeren Zeitraum inaktiv sein werden, ist es möglicherweise erforderlich, auf einen Standby zu wechseln.

Anmerkungen

- Die Console Manager-Befehle **Submit Job** und **Submit Schedule** können nicht auf einem Agenten verwendet werden, der nicht mit seinem Domänenmanager kommunizieren kann.
- Wenn die Verbindung zu einer Standardagentenworkstation zusammenbricht, gibt es keine temporären Wiederherstellungsoptionen, weil Standardagenten sich auf ihren Domänenmanagern befinden. In Netzwerken mit einer großen Anzahl Standardagenten können Sie sich entscheiden, auf einen Domänenmanager im Bereitschaftsmodus zu wechseln.

Einen Domänenmanager im Bereitschaftsmodus einrichten

Die Fehlerbehebung wird erleichtert, wenn Sie auf Netzwerkprobleme vorbereitet sind. Sie sollten speziell folgende Aktionen ausführen:

- Bestimmen Sie einen fehlertoleranten Agenten in der Domäne als Domänenmanager im Bereitschaftsmodus.
- Stellen Sie sicher, dass die Modi 'Vollständiger Status' und 'Abhängigkeiten auflösen' in der Workstationdefinition des Domänenmanagers im Bereitschaftsmodus ausgewählt sind.
- Stellen Sie sicher, dass 'Vollständiger Status' und 'Abhängigkeiten auflösen' für die Domänenmanager (einschließlich des Hauptdomänenmanagers) aktiviert sind. Dies ist wichtig, wenn eine längerfristige Fehlerbehebung erforderlich wird, wobei der Sicherungs-Master eine Symphony-Datei generiert (d. h., Jnextday ausführt). Wenn diese Einträge *nicht* aktiviert sind, wird der bisherige Hauptdomänenmanager nach dem ersten Lauf von Jnextday als normaler fehlertoleranter Agent angezeigt. Während normaler Operationen aktiviert der Job Jnextday automatisch die Markierungen 'Vollständiger Status' und 'Abhängigkeiten auflösen' für den Hauptdomänenmanager, wenn sie nicht bereits aktiviert sind. Wenn der neue Master Jnextday ausführt, erkennt er den bisherigen Hauptdomänenmanager nicht als Sicherungs-Master, wenn diese Markierungen nicht

aktiviert sind. Zu dem Zeitpunkt, zu dem zum bisherigen Master zurückgewechselt wird, hat dieser keine präzise Symphony-Datei. Behandeln Sie die Workstationdefinitionen aller Domänenmanager so, als ob sie Sicherungsdomänenmanagerdefinitionen für den neuen Domänenmanager wären. Dadurch wird echte Fehlertoleranz sichergestellt.

Hauptdomänenmanager im Bereitschaftsmodus

Es ist möglicherweise erforderlich, Dateien zwischen dem Hauptdomänenmanager und dem entsprechenden Domänenmanager im Bereitschaftsmodus zu übertragen. Aus diesem Grund müssen die Computer kompatible Betriebssysteme haben. Kombinieren Sie nicht UNIX- und Windows NT-Computer sowie unter UNIX nicht Computer mit Big-endian und Computer mit Little-endian.

Machen Sie täglich nach der Tagesstartverarbeitung auf dem Hauptdomänenmanager Kopien der Verzeichnisse `Maestrohome\mozart` und `TWShome\..\unison\network` und der Datei `TWShome\Sinfonia`. Falls erforderlich, können die Kopien auf den Hauptdomänenmanager im Bereitschaftsmodus versetzt werden.

Anmerkung: Wenn die Verzeichnisse `Maestrohome/mozart` und `../unison/network` auf dem aktuellen Hauptdomänenmanager unter UNIX einigermaßen statisch sind, können sie im voraus auf den Domänenmanager im Bereitschaftsmodus kopiert werden. Während der normalen Verarbeitung sind die Verzeichnisse verdeckt, wenn Sie die Verzeichnisse des aktuellen Hauptdomänenmanagers auf dem Domänenmanager im Bereitschaftsmodus anhängen. Wenn es erforderlich wird, auf den Domänenmanager im Bereitschaftsmodus zu wechseln, werden die Kopien auf dem Domänenmanager im Bereitschaftsmodus zugänglich gemacht, indem die Verzeichnisse des aktuellen Hauptdomänenmanagers einfach abgehängt werden.

Hinweis zur Netzwerksicherheit

Die Netzwerksicherheit wird gewährleistet, wenn Sie eine Überprüfung von IP-Adressen verwenden. Daraus folgt, dass die Verbindung einer Workstation (mit der Option **Autolink** oder dem Befehl **Link**) möglicherweise fehlschlägt, wenn auf dem Agenten eine alte Symphony-Datei vorhanden ist, die den neuen Domänenmanager nicht enthält. Wenn eine Verbindung fehlschlägt, entfernen Sie die alte Symphony-Datei vom Agenten, und versuchen Sie die Verbindung erneut.

Ausfall eines Domänenmanagers

Der Ausfall eines Domänenmanagers kann als Ergebnis von Netzwerkverbindungsproblemen oder eines Ausfalls des Domänenmanagercomputers selbst auftreten. Der Betrieb ohne einen Domänenmanager hat folgende Auswirkungen:

- Agenten und untergeordnete Domänenmanager können Abhängigkeiten zwischen Workstations nicht auflösen, weil Aktivitätsdatensätze nicht empfangen werden, die durch den Hauptdomänenmanager übertragen werden.
- Standardagenten, deren Host der ausgefallene Domänenmanager ist, können keine Verarbeitung ausführen, weil Sie bei der gesamten Zeitplanung und bei allen Jobstarts vom Domänenmanager abhängig sind.

Wenn Sie erwarten, dass das Problem von kurzer Dauer ist, können Sie es wie im Abschnitt „Netzwerkverbindungsprobleme” auf Seite 109 beschrieben handhaben. Wenn Sie nicht wissen, wie lange das Problem auftritt, oder wenn Sie die normale Verarbeitung der Agenten wiederherstellen wollen, müssen Sie wie im folgenden Abschnitt beschrieben auf einen Domänenmanager im Bereitschaftsmodus wechseln.

Einen Domänenmanager wechseln

Verwenden Sie die folgende Prozedur, wenn der Hauptdomänenmanager kurzfristig ausfällt:

1. Führen Sie Job Scheduling Console aus.
2. Wählen Sie den Master-Connector aus.
3. Wählen Sie **Standardplanlisten** aus.
4. Klicken Sie **Status aller Domänen** an.
5. Klicken Sie **Liste laden** an.
6. Wählen Sie **Manager wechseln...** aus.
7. Geben Sie den Namen des Sicherungsdomänenmanagers an, den Sie verwenden wollen.
8. Klicken Sie **OK** an.

Der Wechsel der Domänenmanager bleibt erhalten, bis Sie eine weitere Operation zum Wechseln der Manager ausführen. Wiederholen Sie diese Prozedur, um zum ursprünglichen Domänenmanager zurückzukehren. Bei einem gewechselten Hauptdomänenmanager müssen Sie diese Prozedur vor der Umstellung auf den nächsten Tag ausführen, wenn Sie nicht erwarten, dass der Hauptdomänenmanager auch am nächsten Tag nicht für die Umstellung auf den nächsten Tag (Zeitplan **final** und Job **Jnextday**) verfügbar ist. Verwenden Sie in diesem Fall die Prozedur im folgenden Abschnitt.

Langfristiger Ausfall des Hauptdomänenmanagers

Verwenden Sie die folgende Prozedur, um auf den Domänenmanager im Bereitschaftsmodus zu wechseln, wenn Sie nicht erwarten, dass der ursprüngliche Master wieder einsatzbereit ist, bevor die Umstellung auf den nächsten neuen Tag (Zeitplan **final** und Job **Jnextday**) ausgeführt wird. Verwenden Sie unter UNIX Schrägstriche in Pfadnamen.

1. Verwenden Sie die Console Manager-Funktion **Stop**, um Tivoli Workload Scheduler auf dem Hauptdomänenmanager und seinem Domänenmanager im Bereitschaftsmodus zu stoppen.
2. Hängen Sie unter UNIX die Verzeichnisse des Masters ab, wenn Sie auf beliebigen Agenten oder Domänenmanagern angehängt sind.
3. Erstellen Sie eine Methode, damit der Domänenmanager im Bereitschaftsmodus auf das Tivoli Workload Scheduler-Dateisystem für UNIX zugreifen kann. Sie können eine der folgenden Methoden verwenden:

   - Kopieren Sie die Verzeichnisse `maestrohome/mozart` und `maestrohome/../unison/network` auf den Domänenmanager im Bereitschaftsmodus. Beachten Sie, dass Sie dies tun müssen, bevor ein vollständiger Systemausfall auf dem ursprünglichen Master auftritt.
   - Richten Sie ein anhängbares Dateisystem ein. Sie sollten dieses Dateisystem extern vom Master und Domänenmanager im Bereitschaftsmodus erstellen, für den Fall, dass auf dem ursprünglichen Master ein vollständiger Systemausfall auftritt.

     Stellen Sie sicher, dass die Maestro-Verzeichnisse `maestrohome/mozart` und `maestrohome/../unison/network` auf dem System, auf dem sie sich befinden, über einen Eintrag in der Datei `etc/exports` angehängt werden können. Hängen Sie die Dateisysteme auf dem Domänenmanager im Bereitschaftsmodus an.

4. Editieren Sie die Datei `maestrohome\mozart\globalopts` auf dem Domänenmanager im Bereitschaftsmodus, und ändern Sie die globale Option **master** in den Workstationnamen des Domänenmanagers im Bereitschaftsmodus.
5. Verwenden Sie Composer auf dem Domänenmanager im Bereitschaftsmodus, um alle wichtigen Jobnetze zu ändern, die auf dem Hauptdomänenmanager ausgeführt werden, wie z. B. der Zeitplan **final**. Ändern Sie jeweils den Workstationnamen in den Namen des Domänenmanagers im Bereitschaftsmodus.
6. Hängen Sie, falls erforderlich, die Verzeichnisse des Domänenmanagers im Bereitschaftsmodus auf Agenten und Domänenmanagern an.
7. Verwenden Sie die Console Manager-Funktion zum Wechseln des Managers, um zum Sicherungs-Master zu wechseln. Siehe „Einen Domänenmanager wechseln” auf Seite 111.

C

Netzwerkbetrieb mit MPE

Dieser Anhang beschreibt die Konfigurationsanforderungen und die Verarbeitung in Tivoli Workload Scheduler/Maestro-Netzwerken, die sich aus einer Kombination aus MPE-, UNIX- und Windows NT-Computern zusammensetzen. Diese Netzwerke bestehen entsprechend der Standardhierarchie von Tivoli Workload Scheduler aus Master, fehlertoleranten Agenten (Slave) und Standardagenten, die folgende Merkmale aufweisen:

- Die Master-Workstation kann entweder ein MPE-, ein UNIX- oder ein Windows NT-Computer sein. Sie ist der Management-Hub in einem Netzwerk. Auf ihm befinden sich die Master-Plandateien von Maestro und er führt für das Netzwerk alle Nachbereitungs- (Tagesende) und Vorbereitungsprozesse (Tagesstart) aus. Wenn ein neuer Arbeitstag beginnt, verarbeitet der Master seine eigenen Zeitpläne sowie die Pläne von Standardagenten-CPUs. Er startet seine eigenen Jobs und sendet Startanforderungen zur Jobausführung an die Standardagenten. Bei der Verwendung von UNIX- oder NT-Mastern wird der Master auch Hauptdomänenmanager genannt. MPE kann für einen Windows NT-Computer kein Master sein, da auf dem MPE-Produkt keine entsprechende Datenbank USERDATA vorhanden ist, auf der Definitionen für NT-Benutzerkennwörter gespeichert werden können.
- Slave-CPUs bei MPE entsprechen den fehlertoleranten Agenten bei UNIX und Windows NT. Für administrative Funktionen sind sie von der Master-Workstation abhängig. Beim Beginn eines neuen Produktionstages verarbeiten sie ihre eigenen Zeitpläne und starten ihre eigenen Jobs.
- Standardagenten-CPUs sind UNIX- oder Windows NT-Computer. Zu administrativen Zwecken sind sie von der Master-Workstation abhängig. Zu Beginn eines neuen Produktionstages führen sie Jobs erst aus, wenn sie vom Domänenmanager (standardmäßig dem Hauptdomänenmanager) entsprechende Startanforderungen empfangen haben.

Überlegungen zum Netzwerk

Bevor Sie entscheiden, wie Sie ein gemischtes Tivoli Workload Scheduler/Maestro-Netzwerk konfigurieren wollen, sollten Sie diesen Anhang komplett durchlesen. Im Folgenden finden Sie eine Zusammenfassung der wichtigsten Überlegungen:

- Objektdefinitionen und Zeitplanung für Slaves und fehlertolerante Agenten müssen auf dem Hauptdomänenmanager erfolgen, wenn sich die Slaves und fehlertoleranten Agenten nicht auf derselben Plattform wie der Master befinden.
- Wenn ein Hauptdomänenmanager im Bereitschaftsmodus benötigt wird, muss er sich auf derselben Plattform wie der Hauptdomänenmanager befinden.
- Die Sicherheitsüberlegungen unterscheiden sich je nach Plattform: MPE, UNIX und Windows NT.

- Die Hauptdomänenmanager von UNIX und NT unterstützen einen hierarchischen Aufbau, bei dem Sie verschiedene Unterdomänenmanager mit Agenten aller unterstützten Plattformen haben können. Die Master von MPE unterstützen nur eine ”flache” Architektur. MPE-Systeme können in jedem hierarchischen Domänenaufbau als FTAs (Slaves) verwendet werden.
- Eine MPE-Master-CPU kann zu MPE-Slaves sowohl mit NS als auch mit TCP/IP eine Verbindung herstellen. UNIX- oder Windows NT-Master-CPUs verwenden ausschließlich TCP/IP.

Überlegungen zur Zeitplanung

Folgendes müssen Sie beachten:

- Bei einem Tivoli Workload Scheduler/Maestro-Netzwerk, in dem eine MPE-Workstation mit anderen Workstations kombiniert ist, müssen die entsprechenden UNIX-/NT-Workstations im nicht erweiterten Modus laufen (keine Zeitplanungsobjektnamen mit mehr als acht Zeichen möglich).
- Da die Sicherheit eines fehlertoleranten Agenten unter UNIX oder Windows NT vom MPE-Master unabhängig ist, kann der fehlertolerante Agent auch in den Modus ’Vollständiger Status’ gesetzt und zum zentralen Konsolmanagement des Netzwerkes verwendet werden. Dadurch können Benutzer Grafikschnittstellen sowie leistungsfähigere und flexiblere Befehlssätze nutzen. Jedoch unterliegt der fehlertolerante Agent weiterhin Beschränkungen bei einigen Conman-Befehlen. Folgende Einschränkungen bestehen:
  - Jobs können nicht mit dem Befehl CONMAN SUBMIT JOB übergeben werden.
  - Zeitpläne können nicht mit dem Befehl CONMAN SUBMIT SCHEDULE übergeben werden.
  - Jobs können nicht mit den Optionen CONMAN RERUN ;FROM und ;FILE wiederholt werden.
  - Abhängigkeiten von vordefinierten (oder globalen) Eingabeaufforderungen können nicht mit dem Befehl ADDDEP hinzugefügt werden.
  - Mit dem Befehl CONMAN ADDDEP können keine Abhängigkeiten von globalen (vordefinierten) Eingabeaufforderungen hinzugefügt werden.
  - Der Befehl CONMAN DISPLAY kann nicht für Jobs oder Zeitpläne verwendet werden.
- Die Funktion **docommand** ist unter UNIX und Windows NT voll funktionsfähig für Jobdefinitionen und den Befehl **conman submit**. Unter MPE wird diese Funktion nicht unterstützt.
- Mit dem UNIX- und Windows NT-Composer können alle Zeitplanungsobjekte (einschließlich Zeitplänen und Jobs) mit einer einzigen Bearbeitungsdatei eingefügt und verändert werden.
- Beim MPE-Master sind INET-Abhängigkeiten nicht möglich. INET-Abhängigkeiten können sich jedoch auf fehlertolerante Agenten von MPE beziehen, wenn der Domänenmanager unter UNIX oder Windows NT läuft.

Installation

Dieses Handbuch enthält Anweisungen zur Installation von Tivoli Workload Scheduler unter UNIX und Windows NT. Angaben zur Erstinstallation von Maestro für MPE finden Sie in Abschnitt 2 des Handbuchs *Maestro User Guide for the MPE System User*.

Einrichtung und Konfiguration

In diesem Abschnitt werden die notwendigen Schritte zum Einrichten von Tivoli Workload Scheduler/Maestro-Netzwerken beschrieben, nachdem sie mit UNIX-Agenten unter MPE, bzw. mit MPE-Agenten unter UNIX- oder Windows NT-Master-Workstations installiert wurden.

Anmerkung: Nähere Angaben zu speziellen Befehlen und Transaktionen finden Sie in diesem Handbuch und im Handbuch *Maestro User Guide for the MPE System User*.

Einrichten eines UNIX-Agenten mit einem MPE-Master

Führen Sie auf dem MPE-Master mit dem Programm ARRANGER Folgendes aus:

1. Erstellen Sie mit Hilfe der Transaktion 'ACPU' eine CPU-Definition für die UNIX-CPU im Netzwerk.
2. Definieren Sie mit der Transaktion 'CSYS' diese CPU als Master.
3. Definieren Sie mit der Transaktion 'ALNK' folgende Verbindungen:
   a. Master-Master: Verbindung mit TCP/IP-Befehlen definieren
   b. Master-UNIX: Verbindung mit TCP/IP-Befehlen definieren
4. Nachdem die unten stehenden Tasks für den UNIX-Agenten beendet sind, werden die Agenten initialisiert und gestartet, sobald auf dem Hauptdomänenmanager das nächste Mal 'Jnextday' ausgeführt wird.

Führen Sie auf jedem fehlertoleranten Agenten und jedem Standardagenten von UNIX Folgendes durch:

1. Ändern Sie die Dateien 'globalopts' und 'localopts' entsprechend Ihren Anforderungen. Standardwerte, die zu den meisten Installationen passen, werden von den Programmen **customize** und **Setup** eingesetzt.
2. Ändern Sie die Sicherheitsdatei und installieren Sie sie. Wenn Sie wollen, können Sie dies auf einem Computer tun und die Datei dann auf die anderen UNIX-Systeme kopieren.
3. Setzen Sie auf jedem UNIX-Agenten den Startbefehl ab, um den 'netman'-Dämon zu starten (falls dieser noch nicht gestartet wurde).

Einrichten eines MPE-FTA unter UNIX oder Windows NT

Ein Tivoli Workload Scheduler-Netzwerk mit MPE-Computern muss im nicht erweiterten Modus installiert sein. Gehen Sie, nachdem die Software installiert ist, nach den unten stehenden Schritten vor, um jede einzelne CPU im Netzwerk zu konfigurieren.

Gehen Sie auf dem UNIX- oder Windows NT-Master wie folgt vor:

1. Verwenden Sie die grafische Benutzerschnittstelle der JS Console oder den CLI-Composer, und erstellen Sie damit Definitionen für alle Workstations. Die Definitionen enthalten auch Informationen zu Verbindungen (Kombination der Transaktionen 'ACPU' und 'ALNK' des Programms 'Arranger').
2. Ändern Sie die Datei 'globalopts' entsprechend Ihren Anforderungen. Folgende globale Optionen nehmen bei MPE-Slaves die Stelle der CTP1-Parameter von ARRANGER ein. Sie sind in der Datei 'globalopts' nicht standardmäßig vorhanden und müssen gegebenenfalls hinzugefügt werden:

rules mode={yes|no}
Ersetzt den CTP1-Parameter für den vollständigen Steuerungsmodus. Wenn Sie diese Option auf **yes** setzen, müssen Sie auch die Option **batchman schedule** auf **yes** setzen.

all userjobs in userjobs schedule={yes|no}
Ersetzt den CTP1-Parameter, mit dem alle Benutzerjobs in den Zeitplan USERJOBS gestellt werden. Sie müssen diese Option auf **no** setzen, wenn **rules mode** auf **yes** gesetzt ist.

set mpe job pri to zero={yes|no}
Ersetzt den CTP1-Parameter, mit dem für alle Benutzerjobs die Priorität 0 erzwungen wird. Sie müssen diese Option auf **no** setzen, wenn **all userjobs in userjobs schedule** auf **yes** gesetzt ist.

batchman schedule={yes|no}
Ersetzt den CTP1-Parameter, mit dem von Batchman erstellten Zeitplänen die Priorität 10 zugeordnet wird. Dies betrifft auch UNIX- und Windows NT-Workstations.

3. Nachdem die unten stehenden Tasks für den MPE-Agenten beendet sind, werden die Agenten initialisiert und gestartet, sobald auf dem Hauptdomänenmanager das nächste Mal 'Jnextday' ausgeführt wird.

Führen Sie bei jedem MPE-Slave mit dem Programm ARRANGER Folgendes aus:

1. Erstellen Sie mit Hilfe der Transaktion 'ACPU' eine Definition für diese CPU und den Master.
2. Geben Sie mit der Transaktion 'CSYS' diese CPU und den Master an.
3. Definieren Sie mit der Transaktion 'ALNK' eine Slave-Master-Verbindung mit TCP/IP-Befehlen.
4. Definieren Sie mit den Transaktionen 'CTP2' und 'CTP3' Parameter für CONMAN und BATCHMAN.
5. Verwenden Sie **mstream** oder **stream** für den Job JBATCHMN.MAESTRO.CCC, um den NETMAN-Prozess zu aktivieren. Damit der NETMAN-Prozess erfolgreich über TCP/IP gestartet und initialisiert werden kann, muss er auf dem MPE-Slave bereits laufen.

Unterschiede bei der Verwendung verschiedener Plattformen

In diesem Abschnitt werden die Ausnahmen beschrieben, die in einem Netzwerk mit MPE-, UNIX- und Windows NT-Workstations gelten, in dem eine MPE-, UNIX- oder Windows NT-Workstation als Master definiert ist. Zusätzliche Informationen zum Betrieb eines Netzwerks finden Sie im Handbuch *Tivoli Workload Scheduler Referenz*, im Handbuch *Tivoli Workload Scheduler Planung und Installation* und *Maestro User Guide for the MPE System User* (bei ROCsoftware erhältlich).

MPE-Master

Die Programme ″Arranger″ und ″Composer″

- Alle Objektdefinitionen und Zeitpläne für fehlertolerante Agenten und Standardagenten unter UNIX müssen auf dem MPE-Master mit COMPOSER und ARRANGER erstellt werden.
- Das COMPOSER-Programm für MPE unterstützt die Funktion **docommand** in Jobanweisungen nicht für Zeitpläne, die auf fehlertoleranten Agenten und Standardagenten unter UNIX und Windows NT ausgeführt werden.
- Wiederherstellungsoptionen für Jobs unter UNIX und Windows NT müssen mit der Transaktion 'XJOB' des Programms 'ARRANGER' dokumentiert werden.
- In der Zeitplanungssprache von MPE ist **scriptname** kein gültiges Schlüsselwort. Verwenden Sie stattdessen den Befehl **jobfilename**, um Jobskripts für UNIX und Windows NT zu definieren. Beispiel:

  ```
  jobfilename " pfadname" streamlogon benutzer
  ```

Conman

- Die Option **docommand** steht im Befehl **submit** nicht zur Verfügung, um damit Befehle zur Ausführung auf einer UNIX- oder Windows NT-CPU zu übergeben.
- Parameter für Platzhalterzeichen und Auswahlsets für bestimmte 'conman'-Befehle werden von einem MPE-Master nicht unterstützt, können aber von einem UNIX-FTA im Modus 'Vollständiger Status' verwendet werden.

Fehlertolerante Agenten unter UNIX

Composer

Sie können ausschließlich 'Composer' zum Dokumentieren, Ändern und Löschen von Maestro-Parametern (**parms**) verwenden, die auf jeder einzelnen Workstation lokal festgelegt sind.

Conman

Folgende CONMAN-Operationen sind nicht zulässig:

- Übergeben von Jobs mit dem Befehl CONMAN SUBMIT JOB.
- Übergeben von Zeitplänen mit dem Befehl CONMAN SUBMIT SCHEDULE.
- Hinzufügen von Abhängigkeiten von vordefinierten (oder globalen) Eingabeaufforderungen mit dem Befehl ADDDEP.
- Verwenden der Optionen ;FROM und ;FILE im Befehl RERUN.
- Anzeigen von Jobs und Zeitplänen mit dem Befehl DISPLAY.

UNIX- oder Windows NT-Master

Composer

- Alle Objektdefinitionen und Zeitpläne für MPE-Slaves müssen mit der grafischen Benutzerschnittstelle der JS Console oder im CLI-Composer auf dem UNIX- oder Windows NT-Master erstellt werden.
- Bei der automatischen Jobdokumentation akzeptiert die Jobanweisung der Zeitplanungssprache von UNIX und Windows NT sowohl die MPE-spezifischen Schlüsselwörter als auch die Datei- und Benutzernamen von MPE. Beispiel:

  ```
  jobfilename file.grp.acct [streamlogon user.acct[, grp]]
  ```

  oder

  ```
  isuserjob [= jobname] streamlogon user.acct[, grp]
  ```

- Das Schlüsselwort **opens** der Zeitplanungssprache von UNIX und Windows NT akzeptiert Dateinamen von MPE. Beispiel:

  ```
  opens [ cpu#]" pfadname"| file.grp.acct[(qualifikationsmerkmale)] [,...]
  ```

Conman

- Mit dem Befehl 'showjob' werden MPE-spezifische Informationen angezeigt:

  ```
  [userjob]
  >> alias is
  >> run again as
  SKEL
  ```

- Der Befehl **submit docommand** ist nicht gültig, wenn er einer MPE-CPU erteilt wird, obwohl dieser Befehl auf dem UNIX- oder Windows NT-Master ausgeführt werden kann.

MPE-Slaves

Arranger

Die einzigen zulässigen Transaktionen in ARRANGER lauten: CTP2, CTP3, xCPU, xLNK, xPRM, xPAS, xSYS.

Composer

COMPOSER kann nicht erfolgreich ausgeführt werden.

Conman

Folgende CONMAN-Operationen sind nicht zulässig:

- Übergeben von Jobs mit dem Befehl CONMAN SUBMIT JOB.
- Übergeben von Zeitplänen mit dem Befehl CONMAN SUBMIT SCHEDULE.
- Hinzufügen von Abhängigkeiten von vordefinierten (oder globalen) Eingabeaufforderungen mit dem Befehl ADDDEP.
- Verwenden der Optionen ;FROM und ;FILE im Befehl RERUN.
- Anzeigen von Jobs und Zeitplänen mit dem Befehl DISPLAY.

D

Auf Tivoli Workload Scheduler migrieren

Die Informationen in diesem Anhang können bei der Migration der Maestro-Version 6.x auf Tivoli Workload Scheduler von Nutzen sein.

Kurze Beschreibung von Tivoli Management Framework für neue Tivoli-Benutzer

Bei Tivoli Management Framework handelt es sich um ein offenes, objektorientiertes Gerüst mit einer Reihe von Managern, Brokern und Agenten, die der Spezifikation 'Object Management Group/Common Object Request Broker Architecture (OMG/CORBA)' entsprechen. Mit der OMG/CORBA-Technologie werden die hauptsächlichen Unterschiede zwischen den verschiedenen Betriebssystemen für den Benutzer unsichtbar gemacht; dadurch können die Hauptdienste in Objekten zusammengefasst werden, die von verschiedenen Managementanwendungen verwendet werden können. Mit Tivoli Management Framework sind Sie unabhängig von Plattformen und verfügen über eine einheitliche Architektur für alle Anwendungen sowie über viele verschiedene Anwendungsprogrammierschnittstellen (Application Program Interface, API), die von der Desktop Management Task Force (DMTF) und von Open Group (früher X/Open) als Grundlage für ein Systemmanagementgerüst verwendet werden. Zu den Tivoli-APIs gehören übliche Netzwerk- und Systemmanagementdienste, wie z. B. Zeitplanung, Transaktionsunterstützung, Konfigurationsprofile und eine Benutzerfunktion für generische Objektdatenbanken.

Die Basiseinheit der Tivoli Management Framework-Funktionalität ist der Tivoli-Verwaltungsbereich (Tivoli Management Region - TMR). Ein Bereich besteht aus einem Tivoli-Server und den Clients, die vom Server verwaltet werden. Auf dem Tivoli-Server befindet sich die Datenbank für diesen Bereich. Je nach Größe und Anforderungen einer Umgebung kann mehr als ein Bereich definiert werden. In diesem Fall können die Bereiche zur gemeinsamen Benutzung von Informationen und Ressourcen miteinander verbunden werden; sie können aber auch eigenständig bleiben. Ein Administrator mit der entsprechenden Berechtigungsklasse kann alle gemeinsam benutzten Ressourcen in einer Gruppe verbundener Bereiche von einem einzigen System aus so verwalten, als ob sich die Ressourcen in dem Bereich des lokalen Systems befänden. Normalerweise kann ein Tivoli-Server bis zu 200 vollständig verwaltete Knoten unterstützen. Ab Tivoli Management Framework Version 3.6 und höher werden auf dem Tivoli-Server und auf bestimmten verwalteten Knoten neue Dienste eingeführt, durch die die verwalteten Knoten als Gateways für Hunderte von Endpunkten fungieren können. Dadurch vergrößert sich der Wirkungsbereich eines einzelnen Bereichs erheblich.

Mit Tivoli Management Framework verfügen Sie über eine serverbasierte Implementierung eines CORBA-ORB (Object Request Broker) und eines Basisobjektadapters (BOA). Außerdem bietet es entsprechende Objekt-, Management- und Arbeitsoberflächendienste sowie eine Implementierung der von Open Group für Systemmanagementgerüste verwendeten APIs. Der Objekt-Dispatcher (oserv) ist der Hauptbestandteil der Framework-Laufzeitkomponente. Er wird als einzelner Multi-Thread-Prozess implementiert, der auf jedem Tivoli-Client innerhalb eines Bereichs und auf dem Tivoli-Server dieses Bereichs im Hintergrund ausgeführt wird. Der Objekt-Dispatcher besteht aus einem Object Request Broker, dem BOA und den entsprechenden Diensten. Der Objekt-Dispatcher, der auf dem Tivoli-Server ausgeführt wird, bietet zusätzliche Dienste, wie z. B. Sicherheitsfunktionen und die Vererbungsauflösung der Implementierung.

Auf einem von Tivoli verwalteten Knoten wird die gleiche Software wie auf einem Tivoli-Server ausgeführt. Sie können auf einem verwalteten Knoten die Tivoli-Arbeitsoberfläche ausführen und andere von Tivoli verwaltete Ressourcen direkt verwalten. Jeder verwaltete Knoten verfügt über seinen eigenen, immer aktiven **oserv**-Dienst, der mit dem **oserv**-Dienst auf dem Tivoli-Server kommuniziert. Ein verwalteter Knoten verfügt außerdem über seine eigene Clientdatenbank. Der Hauptunterschied zwischen einem Tivoli-Server und einem verwalteten Knoten ist die Größe der Datenbank. Außerdem kann es in einem Tivoli-Verwaltungsbereich nur dann verwaltete Knoten geben, wenn ein Tivoli-Server vorhanden ist.

Hinweise zur Installation

Die TWS-Steuerkomponente ist vollkommen unabhängig von Tivoli Management Framework, da es sich hierbei nicht um eine Tivoli Management Framework-Anwendung handelt. Beim Tivoli Workload Scheduler-Connector jedoch handelt es sich um eine Tivoli Management Framework-Anwendung. Sie benötigen den Connector, wenn Sie die Job Scheduling Console, die neue grafische Benutzerschnittstelle, verwenden möchten. In Tivoli Workload Scheduler Version 8.1 ist die traditionelle GUI von Maestro nicht mehr enthalten. Außerdem benötigen Sie die Job Scheduling Console, wenn Sie die beiden Steuerkomponenten von Tivoli Workload Scheduler for z/OS und Tivoli Workload Scheduler von derselben Benutzerschnittstelle aus betreiben wollen.

Installationsposition für den Client der Job Scheduling Console

Job Scheduling Console bietet über eine echte grafische Benutzerschnittstelle die Funktionalität von Composer und Conman. Sie kann auf verschiedenen UNIX- und Microsoft Windows-Plattformen installiert werden. Dafür müssen weder Tivoli Management Framework noch Tivoli Workload Scheduler auf dem Client installiert sein. Als einzige Vorbedingung muss das Client-System über TCP/IP mit mindestens einem Knoten verbunden sein, auf dem der Connector ausgeführt wird.

Installationsposition für die Connector

Beim Connector handelt es sich um einen Dienst von Tivoli Management Framework, über den die Clients der Job Scheduling Console mit der Tivoli Workload Scheduler-Steuerkomponente kommunizieren. Ein Connector kann auf einem System installiert werden, das gleichzeitig ein Tivoli-Server oder ein verwalteter Knoten ist.

Wenn der Connector in Ihrer Tivoli Workload Scheduler-Domäne installiert werden soll, jedoch keine Bereiche vorhanden sind und Sie keine vollständige Tivoli-Managementumgebung implementieren wollen, sollten Sie Tivoli Management Framework als eindeutigen Bereich (und daher als Tivoli-Server) auf jedem Knoten installieren, auf dem der Connector ausgeführt wird.

Sie können Connector auf Workstations installieren, bei denen es sich nicht um den Hauptdomänenmanager handelt. Dadurch können Sie die Version der Symphony-Datei der betreffenden Workstation einsehen. Dies kann von Bedeutung sein, wenn die Job Scheduling Console zum Management der Datenbank mit lokalen Parametern verwendet bzw. der Befehl **submit** direkt auf der Workstation statt über den Master ausgeführt werden soll. Damit die Connector auf einer Workstation installiert werden können, muss es sich bei dieser um einen verwalteten Knoten oder einen Tivoli-Server im Verwaltungsbereich handeln.

Connector auf FTAs installieren

Bei Benutzern ohne Tivoli-Umgebung:

- Bei der Erstellung eines FTA muss die Tivoli Workload Scheduler-Steuerkomponente vor dem Connector installiert werden.
- Stellen Sie vor Beginn der Installation sicher, dass auf dem FTA Tivoli Management Framework installiert ist und dass das System eine vom Connector unterstützte Plattform hat.
- Befolgen Sie die Installationsanweisungen, die im Kapitel zur Installation des Connector im Handbuch *Tivoli Workload Scheduler Console Benutzerhandbuch* beschrieben werden.

Bei Benutzern mit einer Tivoli-Umgebung:

- Normalerweise befindet sich der Hauptdomänenmanager auf einem verwalteten Knoten. Zunächst müssen Sie die Job Scheduling Services/Connector-Klassen auf dem Tivoli-Server installieren.
- Achten Sie darauf, dass Sie keine Instanz auf einem verwalteten Knoten erstellen, auf dem die Tivoli Workload Scheduler-Steuerkomponente nicht installiert ist.

Für alle Benutzer:

- Bedenken Sie, dass Sie während des Connector-Installationsprozesses dazu aufgefordert werden, den Namen einer Tivoli Workload Scheduler-Instanz einzugeben. Dieser Name wird in der Job Scheduling-Baumstruktur der Job Scheduling Console angezeigt. Um Verwirrung zu vermeiden, sollten Sie einen Namen verwenden, der den Namen des FTA enthält.
- Wenn Sie den Connector auf mehreren FTAs in einem Netzwerk installieren, beachten Sie bitte, dass die Namen der Instanzen innerhalb des Tivoli Workload Scheduler-Netzwerks und des Tivoli-Verwaltungsbereichs eindeutig sein müssen.

Hinweise zum Sicherungs-Master

Wenn Sie den Connector auf dem Sicherungs-Master installieren möchten, keine Bereiche vorhanden sind und Sie keine vollständige Tivoli-Managementumgebung implementieren möchten, ist es ratsam, einen neuen Tivoli-Server auf dem Sicherungs-Master zu installieren.

Wenn Sie auf dem Sicherungs-Master keinen Master, sondern einen anderen Tivoli-Server installieren, müssen Sie dennoch dieselben Einträge wie auf dem Master aktivieren, d. h.:

- Starten
- Prüfungsebene für Pläne und Datenbanken
- Zeitzonenaktivierung

Umgang mit nicht zu Ebene 1 gehörenden Mastern in einem vorhandenen Maestro-Netzwerk

Wenn Ihr Hauptdomänenmanager auf einer Plattform läuft, die die Installation von Tivoli Management Framework und dem Connector nicht unterstützt, und Sie Job Scheduling Console verwenden möchten, können Sie aus den folgenden drei Optionen auswählen:

- Versetzen Sie den Hauptdomänenmanager auf eine der unterstützten Plattformen.
- Erstellen Sie einen Sicherungs-Master auf einer unterstützten Plattform.
- Hängen Sie die Master-Datenbanken über NFS an einen FTA an.

Den Sicherungs-Master versetzen

Gehen Sie folgendermaßen vor, um den Hauptdomänenmanager zu versetzen (von UNIX nach UNIX):

1. Wählen Sie einen vorhandenen FTA im Netzwerk aus, oder erstellen Sie einen. Wenn Sie einen FTA erstellen, definieren Sie ihn mit den Optionen 'Abhängigkeiten auflösen' und 'Vollständiger Status'.
2. Packen Sie auf dem Hauptdomänenmanager die Verzeichnisse *maestro/mozart/** und *maestro/../unison/network/** mit dem Befehl **tar**.
3. Entpacken Sie diese Dateien auf dem FTA in Verzeichnissen mit demselben Namen.
4. Kopieren Sie keine Angaben für *Parameter* oder *parameters.KEY* aus dem Basisverzeichnis, da andernfalls FTA-eigene Parameter überschrieben werden. Erstellen Sie eine Liste der Parameter beider Maschinen, und fügen Sie die benötigten Parameter dem FTA hinzu.
5. Ändern Sie auf dem FTA in der Datei *globalopts* den Namen des Masters in den FTA-Namen.
6. Verwenden Sie auf dem alten Master den Befehl **switchmgr**, um auf den neuen Master zu wechseln.
7. Brechen Sie den vorhandenen Zeitplan **final** der laufenden Tagesproduktion ab.
8. Fügen Sie den Zeitplan **final** dem neuen Master hinzu (verwenden Sie den Befehl

   ```
   composer "modify sched=name-alter-master#final"
   ```

   und ändern Sie die Workstation-ID in der Zeitplanzeile).
9. Nach dem Speichern und Hinzufügen löschen Sie den Zeitplan 'name-alter-master#final' mit folgendem Befehl:

   ```
   composer "delete sched=name-alter-master#final"
   ```
10. Übergeben Sie den neuen Zeitplan **final** an die Tagesproduktion.
11. Fügen Sie auf dem alten Master in der Datei *globalopts* den Namen des neuen Masters ein.

    Anmerkung: Es ist nicht notwendig, alle globalen Optionen für jede Workstation im Netzwerk zu editieren. Die Knoten erkennen den neuen Master bei der Ausführung von Jnextday an, wenn der Master sie initialisiert.

Einen Sicherungs-Master erstellen

Gehen Sie zum Erstellen eines Sicherungs-Masters wie folgt vor:

1. Installieren Sie Tivoli Management Framework. Näheres dazu finden Sie im Handbuch *Tivoli Enterprise Installation Guide* .
2. Installieren Sie die Tivoli Workload Scheduler-Steuerkomponente. Siehe Kapitel 2, „Unter Windows NT installieren” auf Seite 19 oder Kapitel 3, „Unter UNIX installieren” auf Seite 35.
3. Installieren Sie den Connector. Näheres hierzu finden Sie im Kapitel zur Installation des Connector im Handbuch *Tivoli Workload Scheduler Console Benutzerhandbuch*.
4. Passen Sie die Workstation mit den Tivoli Workload Scheduler-Sicherheitseinstellungen und lokalen Optionen an. Siehe Kapitel 4, „Sicherheit definieren” auf Seite 49 und Kapitel 5, „Optionale Anpassungsoptionen” auf Seite 69.
5. Definieren Sie die Workstation im Tivoli Workload Scheduler-Netzwerk. Verwenden Sie entweder den Composer-Befehl **cpuname** oder das Fenster **Workstations erstellen** in der Job Scheduling Console. Vergewissern Sie sich, dass Sie bei der Definition die Optionen ’Abhängigkeiten auflösen’ und ’Vollständiger Status’ aktivieren. Näheres dazu finden Sie in den Handbüchern *Tivoli Workload Scheduler Referenz* oder *Tivoli Workload Scheduler Console Benutzerhandbuch*.

Datenbanken des Hauptdomänenmanagers anhängen

Sie können über NFS die folgenden Master-Datenbanken an einen FTA anhängen, der auf einer unterstützten Plattform läuft:

- */usr/lib/maestro/mozart/globalopts* (die ausführbare Kopie)
- */usr/lib/unison/network/cpudata*

In der Workstationdefinition des FTA müssen die Optionen ’Vollständiger Status’ und ’Abhängigkeiten auflösen’ aktiviert sein.

Bevor Sie die Datenbanken anhängen, stellen Sie sicher, dass das Dateisystem, das die benötigten Verzeichnisse enthält, in die Datei */etc/exports* auf der Master-Workstation aufgenommen wurde. Wenn Sie sich entschließen, die Verfügbarkeit des Dateisystems zu steuern, nehmen Sie die entsprechenden Einträge in der Datei */etc/hosts* oder */etc/netgroup* auf dem Master vor.

Der Mount-Punkt auf dem FTA muss derselbe wie auf dem Master sein. Geben Sie z. B. auf dem FTA Folgendes ein:

cd *twshome*
/etc/mount *master-name***:mozart mozart**
/etc/mount *master-name***:../unison/network ../unison/network**

Damit die Datenbanken automatisch angehängt werden, können Sie die Mounts in die Datei */etc/checklist* eingeben.

Wenn Sie diese Lösung verwenden, denken Sie daran, dass die Parameterdatenbank auf dem FTA nicht die des Masters ist, sondern eine lokale Kopie. Dies ist von Bedeutung, wenn Sie Parameter als Teil der Jobdefinitionen verwenden (im Task- oder Anmeldenamen), weil bei der Ausführung von Jnextday alle Parameter, auf die mit dem Zeichen ^ (Winkelzeichen) in Jobdefinitionen verwiesen wird, aus der Parameterdatenbank auf dem Master extrahiert werden.

Sie haben zwei Möglichkeiten, dieses Problem zu umgehen:

- Erstellen Sie ein Skript, das die Parameterwerte vom FTA zum Master hochlädt und ändert. Führen Sie dieses Skript unmittelbar vor Jnextday aus. Wenn Sie Jnextday davon abhängig machen, wird sichergestellt, dass die Parameter erfolgreich hochgeladen werden, bevor Jnextday die Produktion für den nächsten Tag aufbaut.
- Versetzen Sie auf der Master-CPU die Parameterdatenbank ins Verzeichnis *mozart*. Stellen Sie eine Verbindung vom Master zum Basisverzeichnis her. Erstellen Sie daraufhin auf dem FTA eine Verbindung von der Parameterdatenbank in *mozart* zu *twshome*.

Wenn Sie die Zeitzonenfunktion in der Job Scheduling Console aktivieren möchten, müssen Sie die lokale Datei *globalopts* auf dem FTA editieren, um den Eintrag *Timezone Enable* zu setzen.

Umgang mit nicht zu Ebene 1 gehörenden Sicherungs-Mastern in einem vorhandenen Maestro-Netzwerk

Sie können Ihren vorhandenen Sicherungs-Master beibehalten, selbst wenn er nicht auf den unterstützten Plattformen (Ebene 1) ausgeführt wird. Bedenken Sie allerdings, dass Sie die Job Scheduling Console in dieser Umgebung nicht verwenden können. Folgen Sie den Anweisungen im vorhergehenden Abschnitt, wenn Sie sich dazu entschließen, Maschinen zu wechseln.

Mehrere Fenster in der JS Console ausführen

Wenn Sie mehr als sieben aktive JS Console-Fenster auf einem Client benötigen, müssen Sie eine weitere Instanz von Job Scheduling Console auf diesem System installieren. Vergewissern Sie sich während der Installation, dass Sie eine eindeutige Position für die zweite Instanz auswählen. Unter Windows NT muss die Position des Direktaufrufs ebenfalls eindeutig sein.

Eine vorhandene Sicherheitsdatei einsatzbereit machen

Um eine vorhandene Sicherheitsstruktur bei der Migration auf Tivoli Workload Scheduler beizubehalten, müssen Sie für jede Ihrer Maestro-Sicherheitsdefinitionen einen Tivoli-Administrator auf dem Tivoli Management Framework-Server erstellen und dann eine Definition für diesen Administrator in der Sicherheitsdatei erstellen. Weitere Details zur Erstellung eines Tivoli-Administrators finden Sie in der Dokumentation zu Tivoli Management Framework. Näheres zum Editieren der Sicherheitsdatei finden Sie unter Kapitel 4, „Sicherheit definieren” auf Seite 49.

Tivoli-Administratoren hinzufügen

Angenommen, Sie möchten Ihr Maestro-Netzwerk auf Tivoli Workload Scheduler migrieren und das System, auf dem sich zurzeit der Hauptdomänenmanager befindet, hat eine der unterstützten Plattformen. Sie möchten den Connector darauf installieren, so dass Sie eine Anzahl von JS Console-Clients verwenden können. In diesem Fall müssen Sie zuerst Tivoli Management Framework installieren. In diesem Beispiel wird davon ausgegangen, dass Sie einen Tivoli-Server installieren.

Die aktuelle Sicherheitsdatei auf Ihrem Master ist die Folgende:

```
###############################################################
# Security File
###############################################################
# (1)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON THE
#      MASTER DOMAIN MANAGER
user mastersm  cpu=$master + logon=maestro,root
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job                           access=@
schedule                      access=@
resource                      access=@
prompt                        access=@
file                          access=@
calendar                      access=@
cpu                           access=@
parameter  name=@ ~ name=r@   access=@
userobj    cpu=@ + logon=@    access=@
end
###############################################################
# (2)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON ANY
#      WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER.
user sm  logon=maestro,root
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu     access=@
schedule      cpu=$thiscpu     access=@
resource      cpu=$thiscpu     access=@
prompt                         access=@
file                           access=@
calendar                       access=@
cpu           cpu=$thiscpu     access=@
parameter     cpu=$thiscpu
              ~ name=r@        access=@
end
###############################################################
```

Angenommen, diese beiden Benutzer sollen die JS Console verwenden. Nach der Installation der TMF-Software erstellen Sie auf der Tivoli-Arbeitsoberfläche einen zusätzlichen Tivoli-Administrator (neben dem Standardadministrator, der vom Installationsprozess erstellt wird) für jeden Maestro-Benutzer. Nennen Sie den einen **mastersm** und den anderen **sm**. Fügen Sie dann die entsprechenden Definitionen hinzu, so dass die Sicherheitsdatei folgendermaßen aussieht:

```
###############################################################
# Security File
###############################################################
# (1)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON THE
#      MASTER DOMAIN MANAGER
user mastersm  cpu=$master + logon=maestro,root
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job                           access=@
schedule                      access=@
resource                      access=@
prompt                        access=@
file                          access=@
calendar                      access=@
cpu                           access=@
parameter  name=@ ~ name=r@   access=@
userobj    cpu=@ + logon=@    access=@
end
###############################################################
```

```
# (2)  TIVOLI ADMINISTATOR DEFINITION FOR MAESTRO OR ROOT USERS
LOGGED IN ON THE
#      MASTER DOMAIN MANAGER
user mastersm  cpu=$framework + logon=mastersm
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job                           access=@
schedule                      access=@
resource                      access=@
prompt                        access=@
file                          access=@
calendar                      access=@
cpu                           access=@
parameter  name=@ ~ name=r@   access=@
userobj    cpu=@ + logon=@    access=@
end
############################################################
############################################################
# (3)  APPLIES TO MAESTRO OR ROOT USERS LOGGED IN ON ANY
#      WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER.
user sm  logon=maestro,root
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu     access=@
schedule      cpu=$thiscpu     access=@
resource      cpu=$thiscpu     access=@
prompt                         access=@
file                           access=@
calendar                       access=@
cpu           cpu=$thiscpu     access=@
parameter     cpu=$thiscpu
              ~ name=r@        access=@
end
#############################################
################
 (4)  TIVOLI ADMINISTRATOR DEFINITION FOR MAESTRO OR
 ROOT USERS LOGGED IN ON ANY
#      WORKSTATION OTHER THAN THE MASTER DOMAIN MANAGER.
user sm  cpu=$framework + logon=sm
begin
#  OBJECT     ATTRIBUTES       ACCESS CAPABILITIES
# ----------  ------------     ----------------------
job           cpu=$thiscpu     access=@
schedule      cpu=$thiscpu     access=@
resource      cpu=$thiscpu     access=@
prompt                         access=@
file                           access=@
calendar                       access=@
cpu           cpu=$thiscpu     access=@
parameter     cpu=$thiscpu
              ~ name=r@        access=@
end
##########################################################
```

Die neue Sicherheitsdatei gewährt den ursprünglichen Benutzern dieselben Berechtigungen auch auf der Job Scheduling Console.

Darüber hinaus können Sie auf dem Tivoli-Server zwei neue Anmeldenamen für den Tivoli-Administrator **mastersm** hinzufügen:

maestro@rome.production.com
maestro@london.production.com

So erhält jeder berechtigte Benutzer, der sich bei *rome* oder *london* als **maestro** anmeldet, die Berechtigungen, die auch **mastersm** hat.

Sicherheit verwalten

Änderungen der Sicherheitsdatei werden wirksam, wenn eines der folgenden Programme von Workload Scheduler gestoppt und neu gestartet wird:

- conman
- composer
- Connector

Sie müssen die Programme einfach nur beenden. Wenn sie das nächste Mal ausgeführt werden, werden die neuen Sicherheitsdefinitionen erkannt. Verwenden Sie bei den Connectoren den Befehl **wmaeutil**, um sie zu stoppen. Sie werden automatisch bei der Aktualisierung einer beliebigen Abfrage in der Job Scheduling Console neu gestartet.

Connector stoppen, um Änderungen zu implementieren

Ein Connector wird automatisch gestartet, wenn ein Befehl von der JS Console eingegeben wird. Sie müssen ihn jedoch manuell stoppen. Unter Windows NT müssen Sie den Connector stoppen, bevor Sie den Befehl **makesec** eingeben. Unter UNIX können Sie ihn vor oder nach dem Befehl **makesec** stoppen.

Verwenden Sie den Befehl **wmaeutil**, um den Connector zu stoppen.

WMAEUTIL ausführen

Gehen Sie zur Ausführung des Befehls 'wmaeutil' wie folgt vor:

1. Richten Sie die Tivoli-Umgebung ein:
   - Von einer UNIX-Befehlszeile aus:
     - Für ksh:
       ```
       . /etc/Tivoli/setup_env.sh
       ```
     - Für csh:
       ```
       source /etc/Tivoli/setup_env.sh
       ```
   - Von einer Windows NT-Befehlszeile aus:
     ```
     %SYSTEMROOT%\system32\drivers\etc\Tivoli\setup_env.cmd
     ```
2. Geben Sie den folgenden Befehl ein:
   ```
   wmaeutil ALL -stop
   ```

Hinweise zu Zeitzonen

Mit der folgenden Methode wird die Implementierung korrekt durchgeführt:

1. Laden Sie Tivoli Workload Scheduler.

   Die Zeitzonenfunktion ist standardmäßig inaktiviert (Eintrag **timezone enable** in *global-options*). Die Datenbank lässt die Angabe von Zeitzonen bei Workstations zu, aber nicht bei *Startzeiten* und *Terminen* innerhalb von Jobnetzen in der Datenbank. Die Planerstellung (Jnextday) ignoriert alle Zeitzonen in der Datenbank. Sie können keine Zeitzonen an irgendeiner Stelle im Plan angeben.

2. Definieren Sie Workstationzeitzonen.

   Stellen Sie die Zeitzone der Master-Workstation, des Sicherungs-Masters und aller FTAs ein, die sich in einer anderen Zeitzone als der Master befinden. Die Datenbank lässt keine Angabe von Zeitzonen bei *Startzeiten* und *Terminen* zu. Sie können derzeit keine Zeitzone in dem Plan angeben, weil der Eintrag **timezone enable** in *globaloptions* immer noch auf NO gesetzt, also inaktiviert ist.

3. Wenn die Zeitzonen der Workstations korrekt eingestellt sind, setzen Sie **timezone enable** in der Datei *globaloptions* auf YES. Durch diese Einstellung und die Definition der Zeitzone auf der Master-Workstation ist das Tivoli Workload Scheduler-Netzwerk in der Lage, die Zeitzonenunterstützung zu nutzen.

   Ab jetzt können alle Benutzer Zeitzonen überall in der Datenbank verwenden, auch wenn sie bis zur nächsten Ausführung von Jnextday warten müssen, bis sie sie auf *Startzeiten* und *Termine* anwenden können. Bis zur Ausführung von Jnextday können sie keine Zeitzonen im Plan verwenden. Bei der nächsten Ausführung von Jnextday werden die Zeitzonen auf den Plan übertragen, und die JS Console und das Ausgabeprogramm ermöglichen die Angabe von Zeitzonen überall im Plan.

4. Verwenden Sie Zeitzonen für *Start*- und *Until*-Zeiten, wenn Sie sie benötigen.

   Jetzt können Sie alle Zeitzonenreferenzen in der Datenbank und im Plan sowohl mit der JS Console als auch in der CLI verwenden.

   Bei fehlertoleranten MPE-Agenten und AS400-LFTAs (Version 6.1) können Sie Zeitzoneninformationen nur über die JS Console angeben.

Benutzervorgaben an andere Benutzer weitergeben (Angepasste Ansichten/Abfragen)

Mit dieser Option, die in der Job Scheduling Console enthalten ist, kann ein neuer Benutzer der JS Console die Benutzervorgaben eines anderen Benutzers kopieren. Sie kann auch dazu verwendet werden, einen Benutzervorgabensatz an mehrere Benutzer weiterzugeben.

Benutzervorgaben werden in der Datei *GlobalPreferences.ser* gespeichert. Die Datei enthält die Namen und Details, einschließlich der Filter, aller Abfragen (oder Listen), die während einer Sitzung gespeichert wurden. Jedes Mal, wenn Sie die JS Console schließen, wird GlobalPreferences mit den Abfragen aktualisiert, die Sie in der Job Scheduling-Baumstruktur gespeichert oder daraus gelöscht haben.

GlobalPreferences wird lokal in einem Benutzerverzeichnis in *...\JSConsole\dat\.tmeconsole* unter Windows oder in *$HOME/.tmeconsole* unter UNIX gespeichert. Das Benutzerverzeichnis hat einen zusammengesetzten Namen, der Folgendes enthält:

- Den Namen des Tivoli-Administrators, der dem Connector zugeordnet ist, auf den der Benutzer zugreift. Darauf folgt ein *at*-Zeichen (@).
- Den Namen des Systems, auf dem der Connector ausgeführt wird, gefolgt von einem *Unterstreichungszeichen* (_).
- Die Ländereinstellungen des Betriebssystems, auf dem der Connector installiert ist. Die ID ändert sich dynamisch, wenn der Benutzer sich an demselben Connector anmeldet und feststellt, dass die Ländereinstellungen geändert sind.

Anmerkung: Die ersten beiden Angaben sind die Namen, die Sie im Startfenster der Job Scheduling Console eingeben, um die JS Console zu starten.

Beispiel: Sie melden sich zum Starten der JS Console zum ersten Mal an der Maschine **fta12** an, auf der der Connector vom Tivoli-Administrator **tws12adm** installiert wurde. Ein Benutzerverzeichnis namens *tws12adm@fta12_en* (wobei *en* für die Ländereinstellung Englisch steht) wird unter dem oben beschriebenen Pfad auf Ihrer Workstation erstellt.

Jedes Mal, wenn Sie sich an einem anderen Connector anmelden, wird ein neues Benutzerverzeichnis hinzugefügt. Und jedes Mal, wenn Sie die JS Console schließen, wird die Datei *GlobalPreferences.ser* in dem Benutzerverzeichnis, das Ihrer Verbindung entspricht, erstellt oder aktualisiert.

Wenn sich ein Benutzer zum ersten Mal an einem Connector anmeldet, wird er in einem Fenster gefragt, ob eine vorhandene Instanz der Datei *GlobalPreferences.ser* auf einem verbundenen Netzlaufwerk oder über eine URL-Adresse geöffnet werden soll. Wenn Sie eine bestimmte Abfragegruppe an Erstbenutzer weitergeben möchten, stellen Sie erst die entsprechende Instanz der Datei *GlobalPreferences.ser* zur Verfügung, und fordern Sie dann die Benutzer auf, den Dateinamen oder die URL-Adresse in dem Fenster anzugeben. Wenn Sie eine Datei mit Vorgaben an vorhandene Benutzer weitergeben möchten, müssen Sie deren eigene Instanzen der Datei *GlobalPreferences.ser* durch die von Ihnen erstellte Datei ersetzen.

Anmerkung: Eine andere Instanz der Datei *GlobalPreferences.ser* befindet sich im Verzeichnis *...\JSConsole\dat\.tmeconsole\TivoliConsole* unter Windows oder in *$HOME/.tmeconsole/TivoliConsole* unter UNIX. Diese Datei enthält Vorgaben, die die Darstellung der JS Console betreffen, und sollte nicht weitergegeben werden.

Erweiterte Datenbanken zurücksetzen

Für die Zurücksetzung einer erweiterten Datenbank gibt es zwei Methoden:

■

Nutzen Sie die folgende Methode, wenn keine Zeitplanungsobjekte nach der Erweiterung der Datenbank erstellt oder geändert wurden. Wenn Zeitplanungsobjekte erstellt wurden, können diese manuell der Datenbank hinzugefügt werden, nachdem diese in den nicht erweiterten Status zurückversetzt wurde. Wurden Änderungen vorgenommen, müssen diese erneut implementiert werden. Führen Sie folgende Schritte aus:

1. Stellen Sie auf dem Master die Sicherungsdateien im Verzeichnis *mozart.old* in den Verzeichnissen *mozart* und *unison/network* wieder her:
   a. Versetzen Sie die folgenden Dateien in das Verzeichnis *unison/network*:
      - cpudata
      - cpudata.key
      - userdata
      - userdata.key
   b. Versetzen Sie alle anderen Dateien in das Verzeichnis *mozart*.
2. Stellen Sie sicher, dass die globale Option **expanded version** auf NO gesetzt ist.
3. Stellen Sie sicher, dass die CPU-Namen in *globalopts* und *localopts* maximal acht Zeichen lang sind.

■

Verwenden Sie die folgende Methode, wenn viele Zeitplanungsobjekte nach der Erweiterung der Datenbank erstellt oder geändert wurden. Verwenden Sie diese Methode auch dann, wenn Sie die erweiterte Datenbank zurücksetzen möchten und die Zeitplanungsobjekte lange Namen haben.

1. Geben Sie die folgenden **Composer**-Befehle ein, um Dateien zu erstellen, die alle Datenbankinformationen für die Zeitplanungsobjekte enthalten. Die Dateien werden nur dann erstellt, wenn diese Objekte in der Datenbank vorhanden sind:

```
create cpu.txt from cpu=@
create jobs.txt from jobs=@#@
create sched.txt from sched=@#@
create cal.txt from calendars
create prompt.txt from prompts
create parms.txt from parms
create param.txt from parameters
```

2. Versetzen Sie die Objektdateien in den Verzeichnissen *mozart* und *unison/mozart* in ein temporäres Verzeichnis. Die Dateien *globalopts* und *runmsgno* bleiben im Verzeichnis *mozart*. Stellen Sie sicher, dass der CPU-Name in der Datei *globalopts* maximal acht Zeichen lang ist und **expanded version** auf NO gesetzt ist. Stellen Sie weiterhin sicher, dass der CPU-Name in der Datei *localopts* maximal acht Zeichen lang ist.

3. Ändern Sie folgende Dateien, die Sie im ersten Schritt erstellt haben:
   - cpu.txt
   - jobs.txt
   - sched.txt

   Legen Sie für alle CPU-, Domänen-, Job-, Zeitplan- und Benutzernamen Namen mit maximal acht Zeichen fest. Prüfen Sie ebenfalls die Länge der Jobdateinamen und Befehle.
4. Benennen Sie folgende Dateien im Verzeichnis *unison/network* um:
   - cpudata
   - cpudata.key
   - userdata
   - userdata.key
5. Geben Sie folgende Composer-Befehle ein, um die neue nicht erweiterte Datenbank mit den Zeitplanungsobjekten zu füllen:

   ```
   add cpu.txt
   add jobs.txt
   add sched.txt
   replace cal.txt
   replace prompt.txt
   replace parms.txt
   replace param.txt
   ```
6. Benennen Sie die Symphony-Datei um, bevor Sie Jnextday ausführen.

Unter Windows NT migrieren

Wenn Sie auf Windows NT-Workstations von Maestro Version 6.x migrieren, müssen Sie sich als **Administrator** anmelden.

Glossar

A

Abhängigkeit. Eine Vorbedingung, die erfüllt sein muss, damit die Ausführung eines Jobs oder Jobnetzes fortgesetzt werden kann. Die maximale Anzahl von Abhängigkeiten, die für einen Job oder ein Jobnetz zulässig sind, ist 40. Die vier von Tivoli Workload Scheduler verwendeten Arten von Abhängigkeiten sind Folgeabhängigkeiten, Ressourcenabhängigkeiten, Dateiabhängigkeiten und Abhängigkeiten von Eingabeaufforderungen.

Ausschließender Laufzyklus. Ein Laufzyklus, der die Tage festlegt, an denen ein Jobnetz nicht ausgeführt werden kann. Ausschließende Laufzyklen haben Vorrang vor einschließenden Laufzyklen.

B

Batchman. Ein Prozess, der zu Beginn jedes Tivoli Workload Scheduler-Verarbeitungstages gestartet wird, um Jobs gemäß den Informationen in der Symphony-Datei zu starten.

Baumstruktursicht. Die Anzeige auf der linken Seite der Job Scheduling Console, die den Tivoli Workload Scheduler-Server, die Gruppen mit Standardlisten und die Gruppen mit vom Benutzer erstellten Listen zeigt.

Begrenzung. Jobbegrenzungen bieten eine Möglichkeit, eine bestimmte Anzahl von Jobsegmenten zuzuordnen, in denen Tivoli Workload Scheduler Jobs starten darf. Eine Jobbegrenzung kann für jedes Jobnetz und für jede Workstation festgelegt werden. Wenn die Workstationjobbegrenzung beispielsweise auf 25 gesetzt wird, dürfen von Tivoli Workload Scheduler nicht mehr als 25 Jobs gleichzeitig auf der Workstation ausgeführt werden.

Benutzer. Für den Benutzernamen, der im Feld 'Anmelden' der Jobdefinition angegeben ist, muss eine entsprechende Benutzerdefinition vorhanden sein. Dies gilt nur für Windows NT. Die Definitionen liefern die Benutzerkennwörter, die zum Starten von Jobs durch Tivoli Workload Scheduler erforderlich sind.

C

Composer. Eine ältere Befehlszeilenanwendung zum Management der Definitionen Ihrer Planungsobjekte in der Datenbank.

Conman. Eine ältere Befehlszeilenanwendung zum Management der Produktionsumgebung. Conman (Console Manager) führt folgende Tasks aus: Starten und Stoppen von Produktionsprozessen, Ändern und Anzeigen von Plänen und Jobs im Plan sowie zum Steuern der Workstationverbindungen in einem Netzwerk.

D

Datenbank. Die Datenbank enthält alle Definitionen, die Sie für Planungsobjekte wie Jobs, Jobnetze, Ressourcen, Workstations usw. erstellt haben. Sie enthält daneben auch weitere wichtige Informationen, darunter Statistiken zur Job- und Jobnetzausführung und Informationen zu der Benutzer-ID, von der ein Objekt erstellt wurde, sowie zum Zeitpunkt der letzten Änderung eines Objekts. Im Gegensatz dazu enthält der Plan nur die Jobs und Jobnetze (einschließlich abhängiger Objekte), deren Ausführung in der Tagesverarbeitung geplant ist.

Dauer. Die Dauer ist der Zeitraum, in dem der Job nach Ihrer Einschätzung beendet wird. In der Zeitanzeige wird die Dauer von Jobs in der Datenbank mit einem hellblauen Balken in der Mitte des Aktivitätsbalkens oder mit einer hellblauen Raute dargestellt.

Dienstprogrammbefehle. Eine Gruppe von Befehlszeilenprogrammen zum Management von Tivoli Workload Scheduler.

Domäne. Eine benannte Gruppe von Tivoli Workload Scheduler-Workstations, die aus einem oder mehreren Agenten und einem Domänenmanager besteht, der als Management-Hub dient. Zu allen Domänen mit Ausnahme der Hauptdomäne gibt es eine übergeordnete Domäne.

Domänenmanager. Der Management-Hub in einer Tivoli Workload Scheduler-Domäne. Die Kommunikation der Agenten in der Domäne wird vollständig über den Domänenmanager geleitet.

E

Einfacher Laufzyklus. Eine Anzahl benutzerdefinierter Tage, an denen ein Jobnetz ausgeführt wird. Ein einfacher Laufzyklus wird für ein bestimmtes Jobnetz festgelegt und kann nicht von mehreren Jobnetzen verwendet werden. Für weitere Informationen siehe *Laufzyklus*.

Eingabeaufforderung. Eingabeaufforderungen können als Abhängigkeiten für Jobs und Jobnetze verwendet werden. Eine Eingabeaufforderung muss bestätigt werden, damit der davon abhängige Job bzw. das davon abhängige Jobnetz gestartet wird. Es gibt zwei Arten von Eingabeaufforderungen: vordefinierte und Ad-hoc-Eingabeaufforderungen. Eine Ad-hoc-Eingabeaufforderung ist in den Merkmalen eines Jobs oder Jobnetzes definiert und für den betreffenden Job bzw. das betreffende Jobnetz eindeutig. Eine vordefinierte Eingabeaufforderung ist in der Tivoli Workload Scheduler-Datenbank definiert und kann von allen Jobs oder Jobnetzen verwendet werden.

Einschließender Laufzyklus. Ein Laufzyklus, der die Tage festlegt, an denen ein Jobnetz zur Ausführung terminiert ist. Ausschließende Laufzyklen haben Vorrang vor einschließenden Laufzyklen.

Erweiterte Datenbank. Erweiterte Datenbanken unterstützen längere Namen für Datenbankobjekte wie Jobs, Jobnetze, Workstations, Domänen und Benutzer. Erweiterte Datenbanken werden mit dem Befehl 'dbexpand' oder als eine Option während

der Installation konfiguriert. Erweitern Sie die Datenbank erst dann, wenn Sie die Bedeutung und die Auswirkungen dieses Befehls kennen.

Erweiterter Agent. Erweiterte Agenten werden verwendet, um die Jobsteuerungsfunktionen von Tivoli Workload Scheduler in andere Betriebssysteme (wie MVS) und Anwendungen (wie Oracle-Anwendungen, Peoplesoft und Baan) zu integrieren. Erweiterte Agenten verwenden Skripte, auch Zugriffsmethoden genannt, um mit externen Systemen zu kommunizieren.

Externer Job. Ein Job aus einem Jobnetz, der ein vorangegangener Job für einen Job aus einem anderen Jobnetz ist. Ein externer Job wird mit einem Platzhaltersymbol in der Diagrammanzeige des Jobnetzes dargestellt.

F

Fehlertoleranter Agent. Eine Agentenworkstation im Tivoli Workload Scheduler-Netzwerk, die lokale Abhängigkeiten auflösen und zugehörige Jobs ohne Domänenmanager starten kann.

Folgeabhängigkeit. Eine Abhängigkeit, bei der ein Job oder Jobnetz erst ausgeführt werden kann, nachdem andere Jobs oder Jobnetze erfolgreich beendet wurden.

Frühestmögliche Startzeit. Die Uhrzeit, vor der der Job oder das Jobnetz nicht starten kann. Die frühestmögliche Startzeit ist eine Schätzung, die auf früheren Erfahrungen mit dem Ausführen des Jobs oder Jobnetzes basiert. Der Job oder das Jobnetz kann jedoch nach der von Ihnen angegebenen Uhrzeit starten, solange alle anderen Abhängigkeiten erfüllt sind. In der Zeitanzeige wird die Startzeit durch den Anfang (linker Rand) des marineblauen Aktivitätsbalkens dargestellt. Für Job-Instanzen wird die von Tivoli Workload Scheduler berechnete Startzeit mit einem hellblauen Balken dargestellt. Siehe auch “Tatsächliche Startzeit” und “Geplante Startzeit”.

G

Geplante Startzeit. Die von Tivoli Workload Scheduler geschätzte Zeit, zu der eine Job-Instanz gestartet wird. Diese Schätzung basiert auf den Startzeiten früherer Ausführungen.

Globale Optionen. Optionen, die auf alle Workstations eines Tivoli Workload Scheduler-Netzwerkes angewendet werden. Sie werden in der Datei *globalopts* auf dem Hauptdomänenmanager definiert. Siehe auch “Lokale Optionen”.

Grenze. Die Jobgrenze stellt die Hauptsteuerungsmöglichkeit für die Jobausführung auf einer Workstation dar. Sie ist eine Prioritätsstufe, die ein Job oder Jobnetz übersteigen muss, bevor er/es ausgeführt werden kann. Wenn die Grenze beispielsweise auf 40 gesetzt wird, wird verhindert, dass Jobs mit einer Priorität von 40 oder geringer gestartet werden.

H

Hauptdomänenmanager. Die Workstation, die die Dateien verwaltet, die verwendet werden, um Steuerungsobjekte in einem Tivoli Workload Scheduler-Netzwerk zu dokumentieren. Sie erstellt den Plan beim Start jedes Tages und führt sämtliche Protokollierungs- und Berichtsfunktionen für das Netzwerk aus.

Host. Eine Workload Scheduler-Workstation, die für erweiterte Agenten erforderlich ist. Es kann sich um eine beliebige Tivoli Workload Scheduler-Workstation handeln, mit Ausnahme eines anderen erweiterten Agenten.

I

INET-Abhängigkeiten. Eine Abhängigkeit zwischen Jobs oder Jobnetzen in separaten Tivoli Workload Scheduler-Netzwerken. Siehe auch *Netzwerkagent*.

INET-Job/-Jobnetz. Jobs oder Jobnetze in einem fernen TWS-Netzwerk, die als Vorgänger für Jobs oder Jobnetze im lokalen Netzwerk definiert sind (INET = Internetwork). Ein INET-Job wird mit einem Platzhaltersymbol in der Diagrammanzeige des Jobnetzes dargestellt. Siehe auch *Netzwerkagent*.

Interaktive Jobs. Ein Job, der interaktiv auf einer Windows NT-Arbeitsoberfläche ausgeführt wird.

Interner Status. Gibt den aktuellen Status von Jobs und Jobnetzen in der Tivoli Workload Scheduler-Steuerkomponente an. Der interne Status ist in Tivoli Workload Scheduler eindeutig. Siehe auch Status.

J

Jnextday-Job. Die Vor- und Nachbereitung kann vollständig automatisiert werden. Zu diesem Zweck muss ein Job terminiert werden, der am Ende eines jeden Tages ausgeführt wird. Der Job **Jnextday** führt Folgendes aus: Verarbeitung des Folgetages festlegen (in der Symphony-Datei enthalten), Berichte drucken, nicht beendete Jobnetze weiterleiten und Tivoli Workload Scheduler stoppen und erneut starten.

Job. Eine Arbeitseinheit, die auf einer Workstation verarbeitet wird. Die Jobdefinition umfasst einen eindeutigen Jobnamen, der in der Tivoli Workload Scheduler-Datenbank zusammen mit weiteren, zur Ausführung eines Jobs erforderlichen Informationen gespeichert ist. Wenn Sie einen Job einem Jobnetz hinzufügen, können Sie auch seine Abhängigkeiten und Zeitbegrenzungen, wie zum Beispiel die geschätzte Startzeit und den Termin definieren.

Job-Instanz. Ein Job, der im Plan für ein spezifisches Bearbeitungsdatum terminiert wurde. Siehe auch “Job”.

Jobnetz. Eine Liste mit Jobs, die als eine Einheit ausgeführt werden (etwa eine wöchentliche Datensicherungsanwendung), sowie Zeitpunkte, Prioritäten und andere Abhängigkeiten, die die genaue Reihenfolge der Jobausführung bestimmen.

Jobnetz ’final’. Das letzte Jobnetz, das an einem Produktionstag ausgeführt wird. Es enthält einen Job, der die Skriptdatei ’Jnextday’ ausführt.

Jobnetz-Instanz. Ein Jobnetz, das im Plan für ein bestimmtes Bearbeitungsdatum terminiert wurde. Siehe auch *Jobnetz*.

Jobstatus. Siehe “Status”.

K

Kalender. Ein Kalender ist ein definiertes Objekt in Tivoli Workload Scheduler, das eine Liste mit Plandaten enthält. Weil es ein eindeutiges, in der Datenbank definiertes Objekt ist, kann es mehreren Jobnetzen zugeordnet werden. Die Zuordnung eines Kalenders zu einem Jobnetz bewirkt, dass das Jobnetz an den im Kalender angegebenen Tagen ausgeführt wird. Beachten Sie, dass ein Kalender als einschließender oder ausschließender Laufzyklus verwendet werden kann.

L

Laufzyklus. Ein Laufzyklus gibt die Tage an, für die die Ausführung des Jobnetzes terminiert wird. In Tivoli Workload Scheduler gibt es drei Arten von Laufzyklen, die Sie für ein Jobnetz angeben können: den einfachen Laufzyklus, den wöchentlichen Laufzyklus und den Kalenderlaufzyklus (dieser wird in der Regel als Kalender bezeichnet). Beachten Sie, dass jeder Laufzyklustyp einschließend oder ausschließend sein kann. Das bedeutet, dass jeder Laufzyklus die Tage definieren kann, für die ein Jobnetz im Produktionszyklus enthalten oder aus dem Zyklus ausgeschlossen ist. Wenn Sie mehrere Laufzyklen für ein Jobnetz definieren und einschließende und ausschließende Laufzyklen dieselben Tage angeben, hat der ausschließende Laufzyklus Vorrang vor dem einschließenden Zyklus.

Liste. Eine Liste zeigt Jobsteuerungsobjekte an. Sie müssen separate Listen für jedes Jobsteuerungsobjekt erstellen. Für jedes Jobsteuerungsobjekt gibt es zwei Listenarten: eine Definitionsliste in der Datenbank und eine Instanzliste im Plan.

Lokale Optionen. Optionen, die nur auf die Workstation, auf der sie definiert wurden, Anwendung finden. Sie werden in der Datei *localopts* auf jeder einzelnen Workstation eines Tivoli Workload Scheduler-Netzwerkes definiert. Siehe auch “Globale Optionen”.

N

Nachfolgender Job. Ein Job, der erst gestartet werden kann, wenn alle Jobs, von denen er abhängig ist, vollständig ausgeführt wurden.

Netzwerkagent. Eine Art von erweitertem Agenten, der verwendet wird, um Abhängigkeiten zwischen Jobs und Jobnetzen in separaten Tivoli Workload Scheduler-Netzwerken zu erstellen. Siehe auch “INET-Abhängigkeit”.

P

Parameter. Parameter können verwendet werden, um Werte in Ihren Jobs und Jobnetzen zu einzusetzen. Wenn ein Parameter in einem Jobskript verwendet wird, wird der Wert zur Laufzeit eingesetzt. In diesem Fall muss der Parameter auf der Workstation definiert sein, auf der er verwendet wird. Parameter können beim Erstellen von Jobskripts für erweiterte Agenten nicht verwendet werden.

Plan. Eine Prozedur, die alle Jobplanungsaktivitäten enthält, die für einen Zeitraum von einem Tag geplant sind. In Tivoli Workload Scheduler wird der Produktionsplan alle 24 Stunden erstellt und setzt sich aus allen Jobs, Jobnetzen und Abhängigkeitsobjekten zusammen, deren Ausführung für diesen Tag terminiert ist. Die Jobnetze, für die Sie Laufzyklen erstellt haben, werden automatisch terminiert und in den Plan eingeschlossen. Mit fortschreitendem Produktionszyklus werden die Jobs und Jobnetze des Planes entsprechend ihren zeitlichen Einschränkungen und anderen Abhängigkeiten ausgeführt. Jobs oder Jobnetze, die nicht erfolgreich ausgeführt werden können, werden in den Plan für den nächsten Tag übertragen.

Platzhalterzeichen. Bei Tivoli Workload Scheduler sind folgende Platzhalterzeichen zulässig: ? Ersetzt ein alphabetisches Zeichen. % Ersetzt ein numerisches Zeichen. * Ersetzt null oder mehrere alphanumerische Zeichen. Platzhalterzeichen werden im Allgemeinen verwendet, um eine Suche nach einem oder mehreren Objekten in der Datenbank zu präzisieren. Wenn Sie beispielsweise alle Workstations anzeigen möchten, können Sie den Platzhalter Stern (*) eingeben. Wenn Sie hingegen eine Liste mit Workstations von ’standort1’ bis ’standort8’ anzeigen möchten, können Sie ’standort%’ eingeben.

Priorität. Eine Zeitvorgabe im Tivoli Workload Scheduler-Warteschlangensystem zur Ausführung von Jobs und Jobnetzen im Plan. Sie können jedem Job oder Jobnetz eine Prioritätsstufe von **0** bis **101** zuordnen. Bei einer Priorität von **0** findet keine Ausführung statt.

R

Ressource. Ressourcen können entweder physische oder logische Ressourcen auf Ihrem System darstellen. Nachdem sie in der Tivoli Workload Scheduler-Datenbank definiert wurden, können sie als Abhängigkeiten für Jobs und Jobnetze verwendet werden. Sie können z. B. eine Ressource mit dem Namen ’Bänder’ mit einem Einheitenwert von 2 definieren. Definieren Sie anschließend Jobs, die zwei verfügbare Bandlaufwerke als Abhängigkeit erfordern. Jobs mit dieser Abhängigkeit können nicht gleichzeitig ausgeführt werden, da die Ressource “Bänder” jedes Mal, wenn ein Job ausgeführt wird, im Gebrauch ist.

S

Standardlistdatei (stdlist). Eine Standardlistdatei wird für jeden Job erstellt, der von Tivoli Workload Scheduler gestartet wird. Standardlistdateien enthalten Header- und Trailer-Banner, zurückgemeldete Befehle sowie Fehler und Warnungen. Diese Dateien können verwendet werden, um Probleme bei der Jobausführung zu beheben.

Status. Gibt den aktuellen Job- oder Jobnetzstatus innerhalb der Job Scheduling Console an. Der Job Scheduling Console-Status gilt für Tivoli Workload Scheduler und Tivoli Workload Scheduler for z/OS. Siehe auch ’Interner Status’.

Symphony-Datei. Diese Datei enthält die Zeitplanungsinformationen, die vom Produktionssteuerungsprozess (Batchman) für die Ausführung des Planes benötigt werden. Die Datei wird in der Vorbereitungsphase erstellt und geladen. In der Produktionsphase wird sie kontinuierlich aktualisiert, so dass sie den aktuellen Status der Verarbeitung angibt: Arbeit beendet, Arbeit wird ausgeführt, noch ausstehende Arbeit. Zum Verwalten der Verarbeitung kann der Inhalt der Symphony-Datei (Plan) angezeigt und über die Job Scheduling Console geändert werden.

T

Termin. Der letzte Zeitpunkt, zu dem mit der Ausführung eines Jobs oder Jobnetzes begonnen werden kann. Er entspricht der Until-Zeit in der Vorversion (Maestro).

Tivoli Management Framework. Die grundlegende Software, die zur Ausführung der Anwendungen in der Tivoli-Produktgruppe erforderlich ist. Diese Softwareinfrastruktur ermöglicht die Integration von Systemmanagementanwendungen von Tivoli Systems Inc. und der Tivoli-Partner. In Tivoli Management Framework ist Folgendes enthalten: •Object Request Broker (oserv) •Verteilte Objektdatenbank •Grundlegende Verwaltungsfunktionen •Grundlegende Anwendungsdienste •Grundlegende Desktop-Dienste, wie die grafische Benutzerschnittstelle. In einer Tivoli-Umgebung wird Tivoli Management Framework auf jedem Client und Server installiert. Der TMR-Server ist jedoch der einzige Server mit einer vollständigen Objektdatenbank.

Tivoli-Verwaltungsbereich. In einer Tivoli-Umgebung ein Tivoli-Server und die Clients, die er bedient. Eine Organisation kann über mehrere TMRs verfügen. Ein TMR spricht die physische Konnektivität von Ressourcen an, wohingegen ein Richtlinienbereich die logische Ressourcenorganisation anspricht.

V

Vorangegangener Job. Ein Job, der erfolgreich abgeschlossen werden muss, bevor nachfolgende Jobs ausgeführt werden können.

W

Workstation. Eine Workstation ist in der Regel ein einzelner Computer, auf dem Jobs und Jobnetze ausgeführt werden. Workstations sind in der Tivoli Workload Scheduler-Datenbank als eindeutiges Objekt definiert. Eine Workstationdefinition ist für jeden Computer erforderlich, auf dem Jobs oder Jobnetze im Workload Scheduler-Netzwerk ausgeführt werden.

Workstationklasse. Eine Gruppe von Workstations. Es können beliebig viele Workstations in eine Klasse aufgenommen werden. Jobnetze und Jobs können für die Ausführung auf einer bestimmten Workstationklasse zugeorndet werden. Das vereinfacht die Replikation eines Jobs oder Jobnetzes auf viele Workstations.

Wöchentlicher Laufzyklus. Ein Laufzyklus, der die Wochentage angibt, für die die Ausführung des Jobnetzes geplant ist. Ein Jobnetz kann z. B. jeden Montag, Mittwoch und Freitag mit einem wöchentlichen Laufzyklus ausgeführt werden. Ein wöchentlicher Laufzyklus wird für ein bestimmtes Jobnetz festgelegt und kann nicht von mehreren Jobnetzen verwendet werden. Weitere Informationen finden Sie unter ″Laufzyklus″.

X

X-Agent. Siehe ’Erweiterter Agent’.

Z

Zeitbegrenzungen. Zeitbegrenzungen können sowohl für Jobs als auch für Jobnetze angegeben werden. Sie können den Zeitpunkt angeben, zu dem die Ausführung begonnen wird, oder den Zeitpunkt, nach dem eine Ausführung nicht mehr versucht wird. Wenn Sie beides angeben, können Sie einen Zeitrahmen definieren, innerhalb dessen ein Job oder Jobnetz ausgeführt wird. Außerdem können Sie ein Wiederholungsintervall für Jobs festlegen. Zum Beispiel können Sie von Tivoli Workload Scheduler den gleichen Job alle 30 Minuten zwischen 08:30 und 13:30 starten lassen.

Zugriffsmethode. Eine ausführbare Datei, die von erweiterten Agenten verwendet wird, um Verbindungen herzustellen und die Jobausführung auf anderen Betriebssystemen (wie MVS) und Anwendungen (wie Oracle-Anwendungen, Peoplesoft und Baan) zu steuern. Die Zugriffsmethode muss in der Workstationdefinition für den erweiterten Agenten angegeben werden.

Index

Sonderzeichen

/usr/unison/components 46

A

Abhängigkeiten auflösen 122
Aktivieren, Prüfung 92
Aktivieren, Zeitzonenfunktion 90
Anhängen über NFS 122
AT, Schlüsselwort 84, 85

B

Beispiel der Datei ″Security″ 60
Beispielprotokolleinträge, Prüfung 97
Benutzerdefinitionen, Security 49
bm check file, Option 76
bm check status, Option 76
bm check until, Option 76
bm look, Option 76
bm read, Option 76
bm stats, Option 76
bm verbose, Option 76

C

Carryforward 85
carryforward, Option 70, 73
Compiler 83, 85
Connector aktualisieren 92
console, Befehl 82
Cpuname 123

D

database audit level 71
Datum
  an 8
  außer 8
Dumpsec 49

E

E-Mail-Kontakt xii
Erstellen der Datei ″Security″ 49
expanded version 71, 130

F

Fenster
  Workstations erstellen 123
Final 83, 122

G

Gcomposer 120
Gconman 120
Globale Optionen
  Dateibeispiel 72
  Dateischablone 72
  für MPE-Agenten 74
  setzen 69
  Syntax 69
globalopts 122, 130
globalopts, Datei
  Prüffunktion 92
  Zeitzonenfunktion 90
GlobalPreferences.ser 128

H

Hauptdomänenmanager 5
history, Option 71

J

jm job table size, Option 76
jm look, Option 77
jm nice, Option 77
jm no root, Option 77
jm read, Option 77
Jnextday 83
Job 7
job.sched, Datenbank 46
Job Scheduling Console 120
Job Scheduling Console-Client 5
Jobnetz 7

K

Komponentendatei 16
Komponentendatei, Anzeige 17
Konsolnachrichten und Eingabeaufforderungen 81
Kundenunterstützung xiii

L

LD_NO_LIBSTREAMSOCKET 35
Localopts 130
Logman 83
Lokale Optionen
  Dateibeispiel 79
  Dateischablone 79
  setzen 74
  Syntax 74

M

Maestro
  6.x 119
Makesec 49, 127
merge stdlists, Option 77
mm read, Option 77
mm response, Option 77
mm retry link, Option 77
mm sound off, Option 77
mm unlink, Option 78
Mozart 124
Mozart.old 130

N

Nachrichtenebene 82
Netman
  Basisverzeichnis 17
  Lokale Optionen 17
Netman, lokale Optionen 81
Neue Funktionen 89
nm ipvalidate, Option 78
nm mortal 78
nm port, Option 78
nm read, Option 78
nm retry, Option 78

O

Objekt-Dispatcher 120
Onlineveröffentlichungen xi
Oserv 120

P

Parameter 122
Parameters.KEY 122
plan audit level 71
Produktgruppen 16
Prozesseingabeaufforderungen 81
Prozessnachrichten 81
Prüfung
  aktivieren 92
  Beispielprotokolleinträge 97
  Datenbankebene 71
  Header-Format 97
  Planebene 71
  Protokolldateien 92
  Protokollformat 93

R

Reihenfolge der Benutzer, Datei ″Security″ 58
Reihenfolge der Objekte, Datei ″Security″ 58
Rep8 83
Reptr 83
Rückmeldung zu Veröffentlichungen xii
Runmsgno 130

S

Schablonendatei, Security 49
Scheddate 86
Schedulr 83, 85
Security
  Beispiel der Datei 60
  Benutzerdefinitionen 50
  Dateisyntax 50
  Dumpsec 49
  Erstellen der Datei ″Security″ 49
  Makesec 49
  Reihenfolge der Benutzerdefinitionen 58
  Reihenfolge der Objekte 58
  Schablonendatei 49
  Übersicht 49
  UNIX-Superuser 59
  Variablen 59
  verwalten 49
setup_env 127
Setzen, globale Optionen 69
Setzen, lokale Optionen 74
Sfinal 83
Sicherheitsdatei 125
Sicherungs-Master 121
Stageman 83
Start 84, 85
Startzeit 72, 128
stdlist width, Option 78
Submit 121
Switchmgr 122

Symphony 121
Syntax, Datei ″Security″ 50
syslog 81
syslog local, Option 78

T

Tagesstartzeit 72
Termin 84, 128
thiscpu, Option 78
Timezone
  enable, Option 72
Timezone enable 124, 128
Tivoli-Arbeitsoberfläche 120
Tivoli-Kundenunterstützung xiii
Tivoli-Managementserver 119
Tivoli-Verwaltungsbereich 119
TWS Connector 4, 120
  Instanzen 121

U

Übersicht über die Installation 15
UNIX-Superuser, Security 59
UNTIL, Schlüsselwort 84

V

Variablen, Datei ″Security″ 59
Veröffentlichungen bestellen xi
Verwalten der Sicherheit 49
Verwalteter Knoten 119
Vollständiger Status 122
Vorbereitungsberichte 83

W

Wmaeutil 127
wmaeutil, Befehl 92
wr read, Option 79
wr unlink, Option 79

Z

Zeitzone 120
Zeitzonenunterstützung 8

Programmnummer: 5698-WKB

SH12-3010-00